

**TUGAS AKHIR - KS141501**

# **RANCANG BANGUN APLIKASI MONITORING KURIR BERBASIS ANDROID**

**ALEXANDER SIRINGORINGO**  
**NRP 5212 100 110**

**Dosen Pembimbing I**  
**Faizal Johan Atletiko, S.Kom, M.T.**

**JURUSAN SISTEM INFORMASI**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2016**

**FINAL PROJECT - KS141501**

# **DEVELOPMENT OF COURIER MONITORING ANDROID BASED APPLICATION**

**ALEXANDER SIRINGORINGO**  
**NRP 5212 100 110**

**Academic Promoter I**  
**Faizal Johan Atletiko, S.Kom, M.T.**

**INFORMATION SYSTEM DEPARTMENT**  
**Information Technology Faculty**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2016**

## LEMBAR PENGESAHAN

### RANCANG BANGUN APLIKASI MONITORING KURIR BERBASIS ANDROID

#### TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**ALEXANDER SIRINGORINGO**

**5212 100 110**

Surabaya, Juli 2016

**KETUA JURUSAN SISTEM INFORMASI**

**Dr. Ir. Aris Tjahyanto, M.Kom**

**NIP 19650310 199102 1 001**

## LEMBAR PERSETUJUAN

### RANCANG BANGUN APLIKASI MONITORING KURIR BERBASIS ANDROID

#### TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**ALEXANDER SIRINGORINGO**

**5212 100 110**

Disetujui Tim Penguji: Tanggal Ujian : July 2016  
Periode Wisuda : September 2016

**Faizal Johan Atletiko, S.Kom, M.T.**

  
(Pembimbing 1)

**Radityo Prasetyanto.W, S.Kom, M.Kom**

  
(Penguji 1)

**Nur Aini R., S.Kom., M.Sc. Eng.**

  
(Penguji 2)

# **RANCANG BANGUN APLIKASI MONITORING KURIR BERBASIS ANDROID**

**Nama Mahasiswa : Alexander Siringoringo**  
**NRP : 5212 100 110**  
**Jurusan : Sistem Informasi FTIF-ITS**  
**Dosen Pembimbing I: Faizal Johan Atletiko, S.Kom,  
M.T.**

## **ABSTRAK**

*Distribusi barang secara tepat dan merata merupakan hal yang sangat penting bagi kelangsungan proses bisnis suatu badan usaha atau perusahaan. Namun pada saat proses distribusi tersebut dilakukan, suatu permasalahan bisa saja terjadi. Alur pemesanan yang rumit dapat menjadi salah satu permasalahannya seperti yang terjadi pada distribusi obat ke apotek atau rumah sakit. Pemesanan obat dapat dilakukan ketika distributor obat menugasi beberapa salesnya untuk mengunjungi apotek atau rumah sakit dan menawarkan obat yang dijual. Apotek yang akan memesan obat harus terlebih dahulu membuat surat pesanan dan menyerahkannya kepada sales. Semua pesanan dari beberapa apotek atau rumah sakit akan dimasukkan kedalam buku pesanan obat secara manual oleh sales. Buku ini kemudian dikembalikan ke distributor dan setelah melalui beberapa tahapan lagi, pesanan tersebut akan dikirim ke apotek atau rumah sakit pemesan. Permasalahan inilah yang menjadi topik pada Tugas Akhir ini, dimana pada saat pengiriman pesanan kurir bisa saja tidak melakukan tugasnya dengan baik. Untuk itu kurir perlu dipantau untuk mengetahui apakah pesanan telah dikirimkan atau tidak. Dalam Tugas Akhir ini, distributor, apotek dan rumah sakit yang berlokasi di Kota Surabaya dijadikan sebagai batasan penelitian.*

*Adapun tujuan dari Tugas Akhir ini adalah merancang dan membangun aplikasi monitoring kurir. Aplikasi ini berbasis Android dengan menggunakan GPS Location sebagai alat bantu untuk memantau pergerakan kurir dan Android Cloud to Device Messaging untuk mendapatkan notifikasi pesanan pada perangkat android. Penggunaan aplikasi ini akan terdiri menjadi tiga bagian. Aplikasi bagian pertama akan digunakan distributor untuk monitoring lokasi kurir. Aplikasi bagian kedua akan digunakan salesman untuk melakukan pemesanan barang ketika mengunjungi apotek atau rumah sakit. Aplikasi bagian ketiga akan digunakan kurir untuk mengetahui lokasi pengiriman dan barang yang dipesan oleh salesman.*

*Dengan adanya aplikasi monitoring kurir ini, apotek atau rumah sakit yang berlokasi di kota Surabaya dapat melakukan pemesanan obat dengan cepat dan dalam jumlah yang tepat. Aplikasi ini juga dapat membantu distributor dalam memantau lokasi dari kurir, serta mengurangi kesalahan dalam penyampaian barang oleh kurir. Sehingga proses bisnis kedua belah pihak bisa berlangsung dengan lebih baik.*

**Kata Kunci:** Apotek, Kurir Obat, Android C2DM, GPS Location

# **DEVELOPMENT OF COURIER MONITORING BASED ANDROID APPLICATION**

**Student Name** : Alexander Siringoringo  
**NRP** : 5212 100 110  
**Department** : Information System FTIF-ITS  
**Supervisor I** : Faizal Johan Atletiko, S.Kom,  
M.T.

## **ABSTRACT**

*Accuracy and precision is two important point which must be fulfilled in product distribution, so the business proses lifecycle of an enterprise or company would run properly. However, at the time of the distribution process performed, a problem could occur. A complicated ordering flow can be one of the problem as happened in the distribution of medicine to pharmacies or hospitals. Medicine can be ordered when distributors assign some of its salesman for visiting pharmacies or hospitals and offer medicines to be sold. Pharmacies who will order must create a letter and handed it to the salesman. All orders from some pharmacies or hospitals will be entered into the order book manually by the salesman. Afterwards order book returned to distributor and after going through several process again, the order will be sent to the pharmacy or hospital. That issue is the topic of this final project, which at the time of delivery of order, courier could be negligent. Therefore courier needs to be monitored to determine whether the order has been shipped or not. In this final project, distributors, pharmacies and hospitals located in the city of Surabaya serve as limitation of the study.*

*The purpose of this final project is to design and build a courier monitoring application. Application is an android based application using GPS Location to trace courier location, and Android Cloud to Device Messaging for push notification*

*feature. The user of this application is divided into three parts. Application in part one will be used by the medicine distributor company. The second one are used by the salesman to make an order while at the customer premises. The last part are used by the courier for order information and customer location.*

*With this courier monitoring application, pharmacies or hospitals located in Surabaya city could order medicine quickly and in the right amount. This application can also help distributors in order to monitor location of the courier. So that the business process of both parties would run properly.*

***Keywords: Pharmacy, medicine courier, Android C2DM, GPS Location***



## DAFTAR ISI

ABSTRAK .....	v
ABSTRACT .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Tugas Akhir .....	4
1.5 Manfaat Kegiatan Tugas Akhir .....	4
1.6 Relevansi Tugas Akhir .....	4
BAB II TINJAUAN PUSTAKA .....	7
2.1 Studi Sebelumnya .....	7
2.1.1 Perancangan dan Implementasi Aplikasi Pelaporan Penjualan dan Stok Berbasis Android (Studi Kasus: Makintech Multimedia Centre Salatiga) .....	7
2.1.2 Sistem Informasi Pemantauan Tenaga Penjual Berbasis Lokasi (Studi Kasus: PT. Surya Mustika Lampung Cabang Baturaja) .....	8
2.2 Dasar Teori .....	9
2.2.1 Apotek .....	9
2.2.2 <i>Global Positioning System (GPS)</i> .....	10
2.2.3 <i>Android Cloud to Device Messaging (C2DM)</i> .....	11
2.2.4 SQLite .....	12
2.2.5 Google Maps API .....	13
BAB III METODOLOGI PENELITIAN .....	15
3.1 Urutan Pelaksanaan .....	15
3.1.1 Analisa Kebutuhan Perangkat Lunak .....	16
3.1.2 Desain Aplikasi .....	16
3.1.3 Pembuatan Daftar Obat .....	16
3.1.4 Pengembangan Aplikasi .....	17

3.1.5 Pengujian Aplikasi .....	17
BAB IV PERANCANGAN .....	19
4.1 Analisis Kebutuhan .....	19
4.1.1 Kebutuhan Fungsional Sistem .....	20
4.1.2 Kebutuhan Non-Fungsional Sistem .....	21
4.2 Desain Sistem .....	21
4.2.1 Aktor-aktor Dalam Sistem .....	22
4.2.2 <i>Use Case</i> .....	23
4.2.3 <i>Use Case Description</i> .....	24
4.2.4 Pembagian Fungsi Aplikasi .....	34
4.2.5 <i>Sequence Diagram</i> .....	35
4.2.6 Daftar <i>Class</i> dan <i>Method</i> .....	40
4.2.7 <i>Desain Interface</i> .....	42
4.3 Desain Database .....	49
4.3.1 Pemetaan Penggunaan Tabel Database .....	54
BAB V IMPLEMENTASI .....	57
5.1 Daftar Obat .....	57
5.2 Lingkungan Implementasi .....	57
5.3 Pembuatan Aplikasi Sales Obat .....	59
5.3.1 Fungsi <i>Login</i> .....	59
5.3.2 Fungsi Buat Pesanan Baru .....	61
5.4 Pembuatan Aplikasi Kurir Obat .....	64
5.4.1 Fungsi <i>Login</i> .....	64
5.4.2 Fungsi Menampilkan Lokasi Pemesan .....	65
5.4.3 Fungsi Foto Dokumentasi .....	68
5.4.4 Fungsi <i>Push Notification</i> .....	71
5.4.5 Fungsi <i>Track</i> Lokasi Kurir .....	73
5.5 Pembuatan Aplikasi Distributor Obat .....	76
5.5.1 Fungsi <i>Login</i> .....	76
5.5.2 Fungsi Pengelola <i>User</i> .....	77
5.5.3 Fungsi Pengelola Pesanan .....	78
5.5.4 Fungsi <i>Track</i> Kurir .....	80
5.6 Pengujian Aplikasi .....	83
5.6.1 Pengujian Fungsional .....	83

5.6.2 Pengujian Kompatibilitas Aplikasi Android .....	84
BAB VI HASIL DAN PEMBAHASAN .....	85
6.1 Hasil .....	85
6.1.1 Pengujian Fungsional Aplikasi Sales Obat .....	85
6.1.2 Pengujian Fungsional Aplikasi Kurir Obat.....	85
6.1.3 Pengujian Fungsional Aplikasi Distributor Obat....	86
6.1.4 Pengujian Kompatibilitas Aplikasi Android .....	87
6.2 Pembahasan.....	91
6.2.1 Pembahasan Pengujian Fungsional.....	91
6.2.2 Pembahasan Pengujian Kompatibilitas Aplikasi Android.....	92
BAB VII KESIMPULAN DAN SARAN .....	93
7.1 Kesimpulan .....	93
7.2 Saran.....	93
DAFTAR PUSTAKA .....	95
BIODATA PENULIS .....	99

## DAFTAR TABEL

Tabel 2.1 Tabel Perbandingan antar Aplikasi .....	9
Tabel 4.1 Kebutuhan fungsional sistem .....	20
Tabel 4.2 Kebutuhan non-fungsional sistem .....	21
Tabel 4.3 Aktor dalam sistem .....	22
Tabel 4.4 Pemetaan FR dan UC .....	23
Tabel 4.5 UCD login “Sales Obat” .....	24
Tabel 4.6 UCD membuat pesanan obat baru .....	25
Tabel 4.7 UCD menyimpan lokasi pemesan .....	25
Tabel 4.8 UCD melihat daftar obat pesanan .....	26
Tabel 4.9 UCD melihat daftar obat pesanan .....	27
Tabel 4.10 UCD melihat lokasi pemesan .....	27
Tabel 4.11 UCD mengupload foto dokumentasi .....	27
Tabel 4.12 UCD logout “Sales Obat” .....	28
Tabel 4.13 UCD login “Kurir Obat” .....	28
Tabel 4.14 UCD logout “Kurir Obat” .....	29
Tabel 4.15 UCD mengelola daftar pesanan .....	29
Tabel 4.16 UCD menambah user aplikasi .....	30
Tabel 4.17 UCD menentukan kurir pesanan .....	31
Tabel 4.18 UCD melihat hasil <i>track</i> lokasi kurir .....	31
Tabel 4.19 UCD melihat foto dokumentasi pengiriman pesanan .....	32
Tabel 4.20 UCD login “Distributor Obat” .....	33
Tabel 4.21 UCD login “Sales Obat” .....	33
Tabel 4.22 UCD menghapus user aplikasi .....	34
Tabel 4.23 Daftar nama <i>class</i> dan <i>method</i> aplikasi sales obat	40
Tabel 4.24 Daftar nama <i>class</i> dan <i>method</i> aplikasi kurir obat	41
Tabel 4.25 Penjelasan Entitas pada CDM .....	50
Tabel 4.26 Relasi antar Entitas CDM .....	52
Tabel 5.1 Spesifikasi Komputer .....	57
Tabel 5.2 Teknologi yang Digunakan .....	58
Tabel 5.3 Spesifikasi <i>Mobile Phone</i> .....	58
Tabel 5.4 Format Test Case .....	83
Tabel 6.1 Hasil uji coba fungsional aplikasi sales obat .....	85
Tabel 6.2 Hasil uji coba fungsional aplikasi kurir obat .....	86
Tabel 6.3 Hasil uji coba fungsional aplikasi distributor obat .....	86

## DAFTAR GAMBAR

Gambar 2.1 Layanan C2DM [11] .....	12
Gambar 3.1 Alur pengerjaan tugas akhir .....	15
Gambar 4.1 <i>Use case diagram</i> sistem .....	24
Gambar 4.2 Fungsi pada aplikasi sales obat.....	35
Gambar 4.3 Fungsi pada aplikasi kurir obat.....	35
Gambar 4.4 Fungsi pada aplikasi distributor obat.....	35
Gambar 4.5 Sequence diagram login .....	36
Gambar 4.6 <i>Sequence diagram</i> pesanan baru .....	37
Gambar 4.7 <i>Sequence diagram</i> lokasi pemesan .....	37
Gambar 4.8 <i>Sequence diagram</i> upload foto dokumentasi.....	38
Gambar 4.9 <i>Sequence diagram</i> track lokasi kurir .....	39
Gambar 4.10 <i>Sequence diagram</i> menambah user .....	39
Gambar 4.11 <i>Sequence diagram</i> penentuan kurir .....	40
Gambar 4.12 Halaman <i>login</i> aplikasi sales obat .....	43
Gambar 4.13 <i>Tab On Progress</i> dan <i>Done</i> aplikasi.....	44
Gambar 4.14 Menu aplikasi dan Halaman Tambah Pesanan .....	45
Gambar 4.15 Desain halaman informasi pesanan .....	46
Gambar 4.16 Halaman foto bukti.....	47
Gambar 4.17 Halaman pengelolaan user.....	48
Gambar 4.18 Halaman pengelolaan pesanan .....	48
Gambar 4.19 Halaman Track Kurir.....	49
Gambar 4.20 Conceptual Data Model aplikasi .....	50
Gambar 4.21 <i>Physical Data Model</i> aplikasi.....	54
Gambar 5.1 Arsitektur pengembangan aplikasi .....	59
Gambar 5.2 Halaman <i>login</i> aplikasi sales obat .....	60
Gambar 5.3 Penggunaan <i>sql prepared statement</i> .....	60
Gambar 5.4 Penggunaan <i>Sharedpreferences</i> pada aplikasi....	61
Gambar 5.5 Pesanan Baru .....	62
Gambar 5.6 Perintah yang dijalankan pada tombol tambah...	62
Gambar 5.7 Kode program pada tombol “Done”.....	63
Gambar 5.8 Halaman informasi pemesan .....	63
Gambar 5.9 Mengubah titik koordinat menjadi teks alamat ..	64
Gambar 5.10 Kode program inisialisasi lokasi google maps ..	65
Gambar 5.11 Kode program untuk membuat rute perjalanan	66

Gambar 5.12 Contoh rute dari posisi kurir ke lokasi pemesan .....	66
Gambar 5.13 Pengecekan status GPS .....	67
Gambar 5.14 <i>Intent</i> memanggil kamera .....	68
Gambar 5.15 <i>ActivityResult</i> untuk menampilkan foto .....	69
Gambar 5.16 Mengubah file foto kedalam <i>bytearray</i> .....	69
Gambar 5.17 <i>Stringrequest</i> dengan metode <i>POST</i> .....	70
Gambar 5.18 Parameter <i>Stringrequest</i> .....	70
Gambar 5.19 Kode program untuk menyimpan foto .....	71
Gambar 5.20 Registrasi kode <i>token</i> .....	72
Gambar 5.21 <i>Receiver service</i> aplikasi .....	72
Gambar 5.22 Pesan notifikasi dari server .....	73
Gambar 5.23 Tampilan rute pada google maps .....	74
Gambar 5.24 <i>Method onlocationchanged</i> .....	74
Gambar 5.25 <i>Class DBHelper</i> untuk mengelola db SQLite .....	75
Gambar 5.26 Pengubahan status sinkronisasi data .....	76
Gambar 5.27 Validasi <i>username</i> dan <i>password</i> .....	76
Gambar 5.28 <i>Form</i> penambahan <i>user</i> .....	77
Gambar 5.29 Penyimpanan <i>user</i> baru kedalam database .....	77
Gambar 5.30 Tampilan penentuan kurir .....	78
Gambar 5.31 Simpan informasi penentuan kurir .....	79
Gambar 5.32 Daftar pesanan dalam proses pengiriman .....	79
Gambar 5.32 Mengambil data pesanan dalam proses pengiriman .....	79
Gambar 5.34 Daftar pesanan selesai .....	80
Gambar 5.34 Mengambil data pesanan selesai .....	80
Gambar 5.36 Tampilan halaman <i>track</i> kurir .....	81
Gambar 5.37 Mengambil data <i>track</i> lokasi .....	82
Gambar 5.38 Inisialisasi google map .....	82
Gambar 5.39 Membuat garis dengan <i>polylines</i> .....	83
Gambar 6.1 Tampilan <i>login</i> aplikasi .....	88
Gambar 6.2 Tampilan daftar pesanan .....	88
Gambar 6.3 Pengisian informasi pemesan .....	89
Gambar 6.4 Rute perjalanan menuju lokasi pemesan .....	90
Gambar 6.5 <i>Upload</i> foto dokumentasi .....	91

## **DAFTAR LAMPIRAN**

Lampiran A Test Case.....	A-1
---------------------------	-----

# **BAB I**

## **PENDAHULUAN**

Pada bab ini, akan dijelaskan tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan Tugas Akhir, dan relevansi serta manfaat kegiatan Tugas Akhir.

### **1.1 Latar Belakang Masalah**

Distribusi barang merupakan proses yang penting karena menyangkut dua belah pihak yang menjalankan proses bisnis masing-masing. Namun pada saat proses distribusi tersebut dilakukan, suatu permasalahan bisa saja terjadi. Alur pemesanan yang rumit dapat menjadi salah satu permasalahannya, hal ini juga sudah dialami pada penelitian yang dilakukan oleh Sari Wijayanti [1] dengan topik distribusi obat ke apotek dan rumah sakit. Berdasarkan penelitian tersebut, distributor obat akan menugasi beberapa salesnya untuk mengunjungi apotek atau rumah sakit dan menawarkan obat yang dijual. Salesman membawa Daftar Persediaan Obat (DPO) dari distributor, kemudian pihak apotek atau rumah sakit akan membuat Surat Pesanan (SP) dan diberikan kepada salesman. Berdasarkan SP, salesman mencatat kembali pesanan yang diterima secara manual pada Buku Pesanan Obat (BPO). Setelah selesai mengunjungi setiap apotek atau rumah sakit, salesman kembali ke kantor (Distributor) untuk menyerahkan SP dan BPO. Setelah melalui beberapa tahapan, salesman akan mengirimkan obat ke apotek atau rumah sakit pemesan. Permasalahan inilah yang menjadi topik pada Tugas Akhir ini, dimana pada saat pengiriman pesanan kurir bisa saja tidak melakukan tugasnya dengan baik. Untuk itu kurir perlu dipantau untuk mengetahui apakah pesanan telah dikirimkan atau tidak. Dalam Tugas Akhir ini, distributor, apotek dan rumah sakit yang berlokasi di Kota Surabaya dijadikan sebagai batasan penelitian.



Permasalahan yang dapat terjadi yakni kesalahan pengiriman obat, jumlah obat yang dikirim tidak sesuai dengan pesanan, dan bahkan kelalaian seperti melewatkan atau tidak melakukan pengiriman obat ke salah satu apotek atau rumah sakit yang memesan. Untuk mengatasi permasalahan tersebut Tugas Akhir ini akan membuat aplikasi berbasis Android yang dapat mempermudah pemesanan barang dan diharapkan bisa mengatasi kelalaian kurir. Aplikasi dikembangkan dengan beberapa fungsi yang dapat mempermudah proses pemesanan obat, serta dilengkapi fitur monitoring kurir yang mengirimkan obat pesanan. Proses pemesanan obat dimudahkan dengan adanya daftar obat yang tersedia pada distributor dimasukkan kedalam database server. Dengan demikian, obat yang akan dipesan dapat dengan mudah dimasukkan kedalam daftar pesanan obat.

Aplikasi akan dibagi menjadi tiga bagian, aplikasi pertama berbasis android yang digunakan oleh salesman untuk melakukan pemesanan. Aplikasi kedua berbasis android yang digunakan oleh kurir untuk mengetahui lokasi pengiriman dan daftar obat pesanan. Aplikasi ketiga berbasis web yang digunakan oleh distributor untuk memantau lokasi kurir yang sedang bertugas mengirimkan pesanan. Ketika salesman melakukan pemesanan, daftar obat pesanan yang dikirim akan diterima oleh aplikasi yang ada pada server, kemudian diproses hingga daftar tersebut masuk ke aplikasi yang digunakan oleh kurir. Daftar pesanan tersebut akan ditampilkan dalam bentuk pesan notification pada perangkat Android yang digunakan kurir. Fitur ini akan memudahkan kurir untuk mengetahui pesanan tanpa perlu membuka aplikasinya terlebih dahulu. Untuk mengetahui lokasi kurir yang sedang mengirim obat, aplikasi dilengkapi dengan GPS Location sehingga koordinat lokasi dapat dikirim secara berkala ke aplikasi Server.

Aplikasi ini dikembangkan dengan basis Android karena penggunaan beberapa fitur seperti push notification untuk menampilkan daftar pesanan, dan juga GPS Location untuk

mengetahui lokasi kurir. Fitur tersebut tidak dapat berjalan maksimal jika dikembangkan dengan aplikasi basis web. Selain itu, data dari eMarketer menunjukkan bahwa penggunaan perangkat mobile di Indonesia pada tahun 2015 mencapai 55,4 juta pengguna dan penetrasinya akan terus meningkat hingga tahun 2019 yang diperkirakan mencapai 92 juta pengguna [2]. Data tersebut menunjukkan bahwa penggunaan perangkat mobile bukan sesuatu yang langka atau jarang ditemukan. Pemilihan platform Android dilakukan karena penggunaannya lebih banyak dibandingkan dengan beberapa platform lainnya seperti iOS, Windows Phone, dan Blackberry. Data dari International Data Corporation (IDC) menunjukkan bahwa Android memiliki market share sebesar 78,7% pada tahun 2013, dan meningkat pada tahun 2014 menjadi 81,5% dari platform lain yakni iOS, Windows Phone, Blackberry, dan lainnya [3]. Berdasarkan informasi tersebut, berarti sebagian besar perangkat mobile yang digunakan masyarakat adalah berjalan pada Platform Android.

## **1.2 Perumusan Masalah**

Berdasarkan penjelasan latar belakang di atas, rumusan masalah yang menjadi fokus utama dalam tugas akhir ini adalah:

1. Bagaimana merancang dan membangun aplikasi monitoring kurir berbasis android?
2. Bagaimana memanfaatkan *GPS Location* dan *Android Cloud to Device Messaging* sebagai alat bantu pada aplikasi monitoring kurir berbasis android?

## **1.3 Batasan Masalah**

Dari permasalahan yang disebutkan di atas, batasan masalah dalam Tugas Akhir ini adalah:

1. Studi kasus yakni apotek berlokasi di Kota Surabaya.

2. Aplikasi android dikembangkan hingga mendukung android API minimal versi 19.
3. Spesifikasi perangkat *mobile* harus memiliki kamera, GPS, dan dapat mengakses internet.
4. Aplikasi dibuat berbasis android dengan menggunakan alat pengembangan Android Studio versi 1.5.1, SQLite, GPS *Location*, dan Android *Cloud to Device Messaging*.

## **1.4 Tujuan Tugas Akhir**

Tujuan dari Tugas Akhir ini adalah merancang dan membangun aplikasi monitoring kurir yang digunakan oleh distributor obat, salesman, dan kurir. Aplikasi ini berbasis Android dengan menggunakan GPS *Location* dan Android *Cloud to Device Messaging* sebagai alat bantu untuk proses monitoring kurir dan notifikasi daftar pesanan barang.

## **1.5 Manfaat Kegiatan Tugas Akhir**

Melalui Tugas Akhir ini diharapkan dapat memberi manfaat yaitu:

1. Membantu distributor memantau kinerja kurir
2. Membantu apotek dalam kemudahan melakukan pemesanan obat
3. Membantu kurir dalam menemukan lokasi apotek dan memberikan pesanan yang tepat

## **1.6 Relevansi Tugas Akhir**

Tugas akhir ini relevan dengan standar tugas akhir untuk jenjang sarjana sistem informasi karena tugas akhir ini bertujuan untuk membuat aplikasi monitoring kurir berbasis android, dimana aplikasi ini dapat membantu perusahaan distributor barang dalam memonitoring kurir serta membantu mempermudah proses pemesanan barang. Tugas akhir ini

berkaitan dengan mata kuliah Interaksi Manusia Komputer, Pemrograman Perangkat Bergerak, Analisa dan Desain Perangkat Lunak dan Konstruksi Pengembangan Perangkat Lunak.

*Halaman ini sengaja dikosongkan*

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini akan dijelaskan mengenai penelitian sebelumnya dan dasar teori yang dijadikan sebagai acuan atau landasan dalam pengerjaan tugas akhir ini. Landasan teori akan memberikan gambaran tentang konsep dan teknologi apa saja yang digunakan atau diterapkan untuk membangun aplikasi pada tugas akhir ini.

#### **2.1 Studi Sebelumnya**

Berikut ini adalah beberapa penelitian terdahulu yang memiliki keterkaitan dengan tugas akhir ini.

##### **2.1.1 Perancangan dan Implementasi Aplikasi Pelaporan Penjualan dan Stok Berbasis Android (Studi Kasus: Makintech Multimedia Centre Salatiga)**

Aplikasi pelaporan penjualan dan stok berbasis android yang dikembangkan oleh Martina Setiana, dkk [4] memiliki tujuan untuk membantu salesman dalam hal pelaporan penjualan dan pengecekan stok barang. Pelaporan penjualan dapat dilakukan langsung oleh salesman melalui aplikasi yang ada di *smartphone* dengan *input* data transaksi dan otomatis langsung dapat diproses oleh admin. Salesman juga dapat melakukan pengecekan stok barang secara langsung melalui aplikasi tersebut.

### **2.1.2 Sistem Informasi Pemantauan Tenaga Penjual Berbasis Lokasi (Studi Kasus: PT. Surya Mustika Lampung Cabang Baturaja)**

Aplikasi sistem informasi pemantauan tenaga penjual berbasis lokasi yang dikembangkan oleh Made Suwastiko, dkk memiliki tujuan untuk membantu tenaga penjual dalam pencatatan penjualan, mengetahui lokasi toko yang akan dikunjungi dan membantu ketua tim dalam mengetahui informasi penjualan produk, serta mengetahui toko yang telah dikunjungi tenaga penjual [5]. Perusahaan yang menjadi studi kasus pada penelitian tersebut merupakan distributor rokok apache yang memiliki banyak tenaga penjual. Dengan sistem tersebut diharapkan dapat meningkatkan produktivitas kinerja tenaga penjual dan mempermudah ketua tim dalam melakukan pemantauan terhadap tenaga penjual.

Sistem informasi tersebut terdiri dari aplikasi android yang digunakan tenaga penjual dan aplikasi web yang digunakan ketua tim. Aplikasi android memiliki beberapa fungsi yang dapat memudahkan tenaga penjual melakukan pendataan toko yang dikunjungi, serta produk yang dibeli oleh toko tersebut. Sedangkan pada aplikasi web memiliki fungsi yang memudahkan ketua tim dalam mengelola tenaga penjual, dan memantau lokasi toko penjualan.

Penelitian yang menghasilkan aplikasi pada pembahasan sebelumnya memiliki perbedaan dengan aplikasi monitoring kurir pada tugas akhir ini. Perbedaan tersebut dapat dilihat pada tabel 2.1.

Tabel 2.1 Tabel Perbandingan antar Aplikasi

<b>Aplikasi</b>	<b>Pengelola Pesanan</b>	<b><i>Tracking</i> posisi salesman/Kurir</b>	<b>Pengelola Salesman/kurir yang bertugas</b>
<b>Aplikasi Pelaporan Penjualan dan Stok</b>	Ya	Tidak	Distributor
<b>Sistem Informasi Pemantauan Tenaka Penjual Berbasis Lokasi</b>	Ya	Tidak	Ketua Tim
<b>Aplikasi <i>Monitoring</i> Kurir</b>	Ya	Ya	Admin Distributor

## 2.2 Dasar Teori

### 2.2.1 Apotek

Apotek merupakan suatu sarana pelayanan kesehatan dalam membantu mewujudkan tercapainya tingkat kesehatan yang optimal bagi masyarakat. Juga sebagai suatu tempat untuk mengabdikan bagi para apoteker dalam melakukan pekerjaan kefarmasian [6]. Menurut Peraturan Pemerintah No.51 tahun 2009 [7], adapun tugas dan fungsi apotek adalah:



1. Tempat pengabdian profesi seorang apoteker yang telah mengucapkan sumpah jabatan apoteker.
2. Sarana yang digunakan untuk melakukan pekerjaan kefarmasian.
3. Sarana yang digunakan untuk memproduksi dan distribusi sediaan farmasi antara lain obat, bahan obat, obat tradisional, kosmetika.
4. Sarana pembuatan dan pengendali mutu sediaan farmasi, pengamanan, pengadaan, penyimpanan, dan pendistribusian atau penyaluran obat, pengelolaan obat, pelayanan obat atas resep dokter, pelayanan informasi obat serta pengembangan obat, bahan obat dan obat tradisional.

Berdasarkan penelitian yang dilakukan oleh Sari Wijayanti [1]. Pemesanan obat dilakukan ketika salesman datang mengunjungi apotek tersebut. salesman membawa Daftar Persediaan Obat (DPO) yang diserahkan distributor, kemudian pihak apotek akan membuat Surat Pesanan (SP) dan diberikan kepada salesman. Berdasarkan SP, salesman mencatat pesanan yang diterima secara manual pada Buku Pesanan Obat (BPO). Setelah salesman selesai mengunjungi apotek, salesman kembali ke Kantor (Distributor) untuk menyerahkan SP. Setelah melalui beberapa tahapan, salesman akan mengirimkan obat ke apotek pemesan.

### ***2.2.2 Global Positioning System (GPS)***

*Global Positioning System* (GPS) merupakan suatu alat atau sistem yang dapat digunakan untuk menginformasikan penggunaanya dimana lokasi pengguna tersebut berada yang berbasis satelit [8]. Data dikirim dari satelit berupa sinyal radio dengan data digital.

Penentuan posisi dengan GPS dilakukan berdasarkan pada pengukuran jarak yang dilakukan ke beberapa satelit sekaligus. Koordinat suatu titik di bumi dapat ditentukan dari hasil

pengukuran ke empat satelit yang dapat ditangkap sinyalnya dengan baik.

Menurut penelitian yang dilakukan oleh Rifqi Andikasani dkk [9], penentuan posisi dengan GPS dibagi menjadi dua metode yakni :

1. Metode absolut atau yang dikenal dengan *point positioning*, adalah cara penentuan posisi dengan hanya berdasarkan pada satu *receiver* saja. Ketelitian posisi dalam beberapa meter (tidak berketelitian tinggi) dan umumnya hanya diperuntukkan bagi keperluan navigasi.
2. Metode relatif atau yang dikenal dengan *differential positioning*, adalah penentuan posisi dengan menggunakan lebih dari satu *receiver*. Satu GPS dipasang pada suatu lokasi dimuka bumi, dan secara terus menerus menerima sinyal dari satelit dalam jangka waktu yang ditentukan dan digunakan sebagai referensi bagi yang lainnya. Posisi dari metode ini memiliki ketelitian yang tinggi (umumnya kurang dari satu meter) dan diaplikasikan untuk keperluan survey geodesi ataupun pemetaan yang memerlukan ketelitian tinggi.

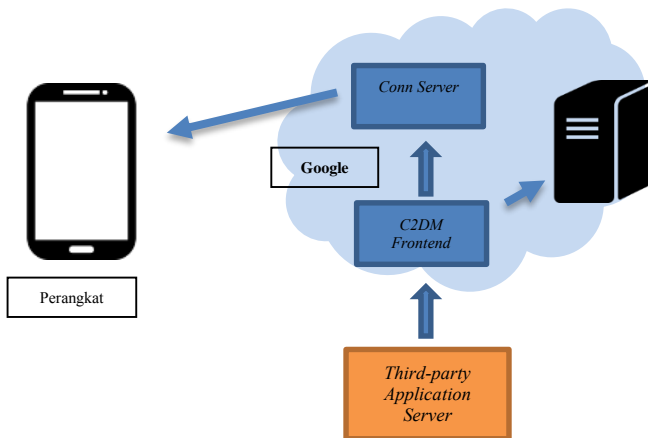
### **2.2.3 Android Cloud to Device Messaging (C2DM)**

Android *Cloud to Device Messaging* adalah layanan yang membantu para pengembang aplikasi untuk mengirimkan data dari server ke aplikasi yang terpasang pada perangkat Android. C2DM ini memiliki mekanisme dimana server dapat memberikan atau mengambil informasi kepada client (aplikasi di perangkat Android). C2DM ini menangani semua bagian dari antrian pesan sampai kepada pengiriman pesan ke aplikasi.

Agar dapat menjalankan layanan C2DM ini dengan baik, maka dibutuhkan beberapa komponen pendukung yakni sebagai berikut [10]:

1. Perangkat Android, yakni perangkat yang menjalankan aplikasi Android yang menggunakan C2DM.
2. *Third-party Application Server*, yakni sebuah server yang berfungsi sebagai bagian dari proses C2DM. Server inilah yang mengirim data ke aplikasi android melalui server C2DM (Google Cloud). Server ini juga harus dilengkapi dengan *back-end* agar bisa mengelola proses dari C2DM yang mengalami masalah.
3. C2DM Server, yakni server Google yang terlibat langsung dalam pengambilan pesan dari *Third-party Application Server* dan mengirimnya ke aplikasi android.

Gambaran layanan C2DM ini diilustrasikan seperti pada gambar 2.1 dibawah.



Gambar 2.1 Layanan C2DM [11]

## 2.2.4 SQLite

SQLite merupakan suatu sistem manajemen basis data relasional yang bersifat ACID – *compliant* (*atomic*, konsistensi basisdata, isolasi, dan durabilitas) [12]. SQLite adalah fasilitas android yang dapat digunakan untuk membuat *database*.

SQLite ini sangat memadai dalam hal sumber daya, karena penggunaan *memory* yang sedikit, cepat, dan ringan.

Pada penelitian ini, SQLite akan digunakan untuk menyimpan data pesanan dan juga koordinat lokasi dari kurir. Dengan penggunaan SQLite ini, update informasi ke server akan dilakukan secara *A-Synchronius*. Hal ini dilakukan jika perangkat tidak memiliki akses internet ketika digunakan.

### 2.2.5 Google Maps API

Google Maps API merupakan sebuah API yang disediakan oleh Google dan dapat digunakan secara gratis untuk menampilkan informasi-informasi geografis dalam bentuk map. Google maps API sangat membantu dalam menerjemahkan informasi geografis seperti titik koordinat ke dalam bentuk grafis yakni map.

Google Maps API digunakan untuk aplikasi web yang akan digunakan oleh distributor untuk memetakan pergerakan posisi kurir dan untuk aplikasi android yang akan digunakan oleh kurir sendiri untuk mengetahui lokasi dari pemesan obat.

Untuk menggunakan Google Maps API kedalam aplikasi web, dapat dilakukan dengan beberapa langkah berikut ini:

1. Memasukkan *script maps* API kedalam kode aplikasi.
2. Mengatur posisi map yang akan tampil pada halaman web.
3. Buat beberapa objek literal untuk menyimpan properti map.
4. Buat fungsi *javascript* untuk inisialisasi map.

Pada aplikasi android, penerapan Google Maps API dilakukan dengan beberapa langkah berikut ini:

1. Registrasi ke halaman web developer google maps untuk mendapatkan *key* agar bisa menggunakan layanan tersebut.

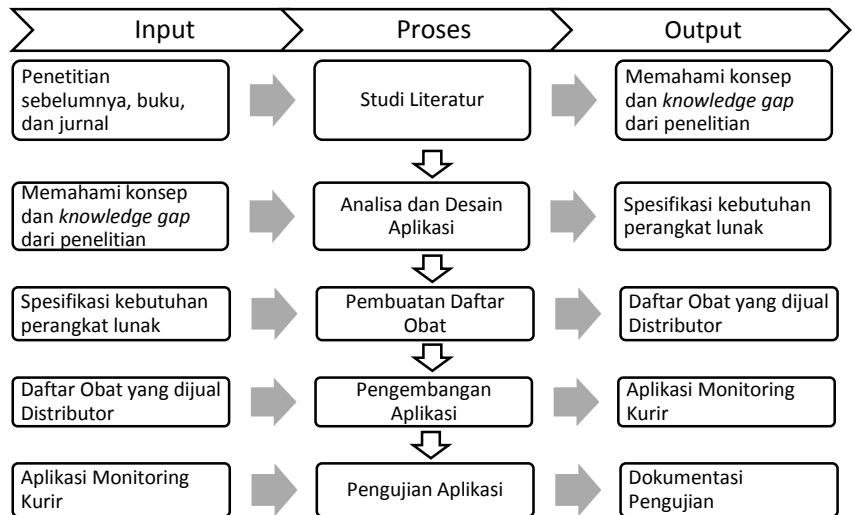
2. Menambahkan *required permission* kedalam manifest aplikasi. Seperti *Location Permission*, dan *Internet Permission*.
3. Menambahkan *Google Play service* kedalam manifest aplikasi.
4. Buat objek untuk menampilkan map didalam *class* aplikasi.

## BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang gambaran metode pengerjaan serta alur pengerjaan tugas akhir ini. Bab ini menjadi acuan pengerjaan tugas akhir sehingga proses pengerjaan menjadi lebih terstruktur dan sistematis.

### 3.1 Urutan Pelaksanaan

Bagian ini menjelaskan metodologi yang digunakan pada tugas akhir ini. Gambar 3.1 merupakan metodologi pengerjaan tugas akhir ini.



Gambar 3.1 Alur pengerjaan tugas akhir

Pengerjaan aplikasi mengikuti alur rancangan pada penelitian Yohanda [13]. Rancangan dimulai dari analisis kebutuhan, desain sistem, desain database, dan desain interface.

### 3.1.1 Analisa Kebutuhan Perangkat Lunak

Pada tahap ini terlebih dahulu dijelaskan bagaimana proses distribusi obat dari distributor ke apotek atau rumah sakit. Kemudian diberikan ide yakni pembuatan aplikasi monitoring kurir ini sebagai alat untuk membantu pada proses distribusi obat yang sudah ada. Dari ide pembuatan aplikasi tersebut kemudian disusun *Functional* dan *Non-Functional Requirement* (FR & non-FR).

### 3.1.2 Desain Aplikasi

Pada tahap ini, rancangan sistem akan dibuat dan didokumentasikan. Rancangan sistem meliputi desain sistem yang didalamnya terdapat penjelasan setiap aktor yang terlibat, pembuatan *usecase* beserta deskripsi setiap *usecase*, pembuatan *sequence diagram*, pembuatan daftar *class* dan *method* yang akan digunakan pada setiap aplikasi, serta pembuatan desain *interface*. Dilanjutkan dengan pembuatan desain database, serta pemetaan penggunaan tabel pada setiap aplikasi. Rancangan aplikasi dibuat berorientasi objek, agar setiap fungsi yang akan dikembangkan bisa dijelaskan dengan baik. Tetapi pada pembuatan aplikasi, pengembangan dilakukan dengan orientasi setiap fungsi harus ada dan dapat berjalan.

### 3.1.3 Pembuatan Daftar Obat

Pada proses ini akan dilakukan pengumpulan data mengenai daftar obat yang dijual oleh distributor. Proses pengumpulan data ini dilakukan dengan penggalan informasi daftar obat pada website perusahaan distributor obat. Daftar obat yang didapatkan akan digunakan dalam memudahkan *input* nama obat ketika melakukan pemesanan.

### 3.1.4 Pengembangan Aplikasi

Pada proses ini aplikasi mulai dikembangkan berdasarkan desain yang telah dibuat sebelumnya. Teknologi yang digunakan untuk membuat aplikasi yakni : Android SDK, GPS Location, C2DM, SQLite, Google Maps API, dan pengembangan web berbasis PHP.

### 3.1.5 Pengujian Aplikasi

Setelah aplikasi selesai dibuat, pengujian aplikasi dilakukan untuk mengecek apakah aplikasi sudah berjalan dengan baik atau belum sesuai dengan FR dan non-FR yang diwujudkan dalam *Requirement Traceability Matrix*. Aplikasi ini juga akan dilakukan *compability test* yaitu pengujian aplikasi dengan menjalankan pada skenario pengujian atau *test case* di berbagai perangkat Android mulai Android API versi 19 hingga Android API versi terbaru (pada penelitian ini Android API versi terbaru adalah 23). Pengujian API Android pada range tersebut dikarenakan fungsi sinkronisasi data yang akan digunakan pada aplikasi dalam Tugas Akhir ini berjalan dengan baik pada *range* API tersebut.



*Halaman ini sengaja dikosongkan*

## **BAB IV**

### **PERANCANGAN**

Pada bab ini, akan dijelaskan perancangan yang akan dilakukan untuk membangun aplikasi pada tugas akhir ini. Aplikasi monitoring kurir ini dirancang dengan beberapa langkah yakni analisis kebutuhan, desain sistem, desain *database*, dan desain *interface* aplikasi sesuai dengan tahapan yang disebutkan sebelumnya.

#### **4.1 Analisis Kebutuhan**

Pada bagian ini, terlebih dahulu dijelaskan bagaimana proses distribusi obat dari perusahaan distributor obat ke apotek atau rumah sakit.

Berdasarkan penelitian yang dilakukan oleh Sari Wijayanti dalam pembuatan aplikasi web mobile pemesanan obat [1], pemesanan obat dilakukan ketika salesman datang mengunjungi apotek atau rumah sakit. Salesman membawa Daftar Persediaan Obat (DPO) dari distributor, kemudian pihak apotek atau rumah sakit akan membuat Surat Pesanan (SP) dan diberikan kepada salesman. Berdasarkan SP, salesman mencatat kembali pesanan yang diterima secara manual pada Buku Pesanan Obat (BPO). Setelah selesai mengunjungi setiap apotek atau rumah sakit, salesman kembali ke kantor (Distributor) untuk menyerahkan SP dan BPO. Setelah melalui beberapa tahapan, salesman akan mengirimkan obat ke apotek atau rumah sakit pemesan.

Berdasarkan proses distribusi dengan cara tersebut, dapat dilihat bahwa selama ini proses pemesanan tersebut tergolong rumit dan membutuhkan waktu yang tidak sedikit. Proses pengiriman obat yang sudah dipesan juga tidak kalah rumit, karena semua pesanan terlebih dahulu dikumpulkan kemudian dikirim ke alamat pemesan. Dengan pendataan pesanan secara

manual didalam kertas, maka besar kemungkinan adanya pesanan yang belum dikirim atau terjadi kesalahan pengiriman.

Oleh karena itu, tugas akhir ini memberikan ide yakni pembuatan aplikasi monitoring kurir yang akan mempermudah proses pemesanan obat. Aplikasi monitoring kurir ini akan dibagi menjadi tiga bagian, yang pertama adalah aplikasi “Sales Obat”. Setiap sales akan memiliki aplikasi tersebut dan melakukan pemesanan obat langsung dari aplikasi ketika berkunjung ke apotek atau rumah sakit. Aplikasi kedua adalah “Kurir Obat”, kurir yang bertugas untuk mengantarkan pesanan akan menggunakan aplikasi ini yang berfungsi untuk menampilkan daftar pesanan beserta informasi pesanan didalamnya. Aplikasi ketiga adalah “Distributor Obat” yang digunakan oleh distributor yang berfungsi untuk mengatur pesanan yang masuk dan menetapkan kurir yang akan mengantarkan pesanan tersebut juga dapat digunakan untuk menampilkan hasil *tracking* lokasi kurir dalam google maps.

#### 4.1.1 Kebutuhan Fungsional Sistem

Berdasarkan skenario alternatif yang dijabarkan sebelumnya, maka disusun kebutuhan fungsional aplikasi monitoring kurir yang sudah disesuaikan dengan proses distribusi obat yang dijelaskan sebelumnya. Kebutuhan fungsional sistem ini tertera pada tabel 4.1.

Tabel 4.1 Kebutuhan fungsional sistem

ID	Kebutuhan Fungsional
FR-01	Sales dapat membuat pesanan obat baru
FR-02	Sales dapat melihat detail pesanan
FR-03	Sales dapat mengelola akun
FR-04	Kurir dapat melihat detail pesanan
FR-05	Kurir dapat mengupload foto dokumentasi
FR-06	Kurir dapat mengelola akun
FR-07	Distributor dapat mengelola pesanan

FR-08	Distributor dapat mengelola user aplikasi
FR-09	Distributor dapat menentukan kurir untuk mengantar pesanan
FR-10	Distributor dapat melihat hasil <i>tracking</i> lokasi kurir
FR-11	Distributor dapat melihat foto dokumentasi pengiriman pesanan.
FR-12	Distributor dapat mengelola akun

#### 4.1.2 Kebutuhan Non-Fungsional Sistem

Dari hasil analisis kebutuhan yang dilakukan sebelumnya, didapatkan kebutuhan non-fungsional yang ada pada tabel 4.3.

Tabel 4.2 Kebutuhan non-fungsional sistem

ID	Kebutuhan Non-Fungsional	Jenis
NFR-01	Sistem dapat diakses 24 jam sehari, 7 hari seminggu	<i>Accessibility</i>
NFR-02	Sistem dapat dijalankan menggunakan sistem operasi Android dengan versi API minimal 19.	<i>Accessibility</i>
NFR-03	Sistem harus bisa menyimpan semua data yang dimasukkan, baik detail obat pesanan, lokasi pemesan, foto dokumentasi, serta hasil <i>track</i> lokasi kurir.	<i>Reliability</i>

#### 4.2 Desain Sistem

Pada tahap desain sistem ini dijelaskan bagaimana aplikasi monitoring kurir ini akan berjalan. Penjelasan dimulai dari aktor-aktor yang ada dalam sistem, pembuatan *use case* beserta deskripsinya, pembuatan *sequence diagram* aplikasi, pembuatan daftar *class* beserta *method* yang akan digunakan pada setiap aplikasi, serta desain *interface* aplikasi.

### 4.2.1 Aktor-aktor Dalam Sistem

Aktor-aktor yang terlibat dalam sistem serta deskripsinya dijelaskan pada tabel 4.3.

Tabel 4.3 Aktor dalam sistem

<b>Aktor</b>	<b>Deskripsi</b>	<b>Aplikasi yang digunakan</b>
Sales	Orang yang bertugas untuk menawarkan obat yang dijual oleh distributor ke apotek atau rumah sakit. Menggunakan aplikasi untuk membuat pesanan baru ketika apotek atau rumah sakit melakukan pemesanan.	Sales Obat
Kurir	Orang yang bertugas untuk mengirim pesanan ke apotek atau rumah sakit pemesan. Menggunakan aplikasi untuk melihat detail pesanan, lokasi pemesan, serta mengupload foto dokumentasi.	Kurir Obat
Distributor	Perusahaan distribusi obat. Menggunakan aplikasi untuk mengelola pesanan, menentukan kurir yang bertugas mengirim pesanan, mengelola user aplikasi, serta melihat hasil <i>track</i> lokasi kurir.	Distributor Obat

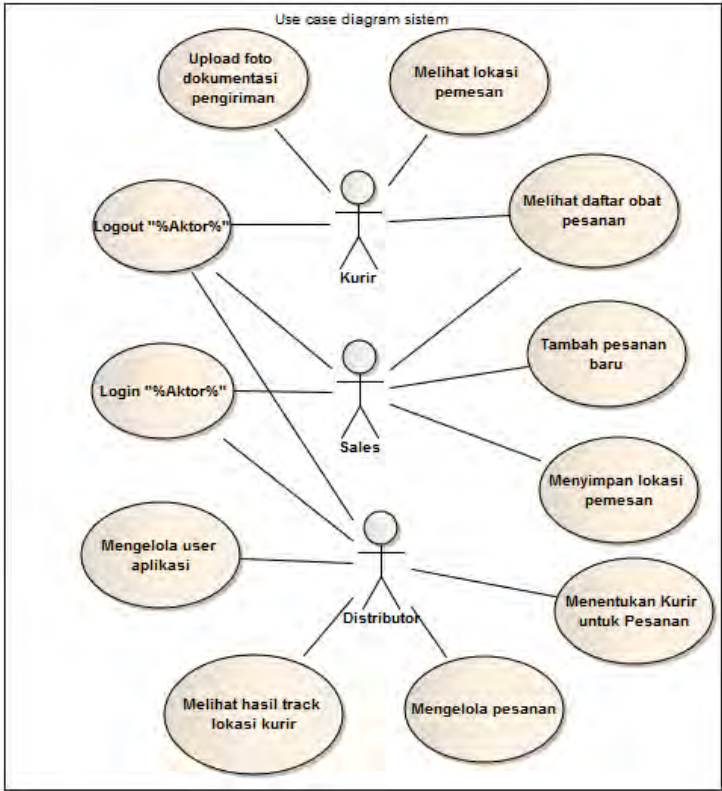
### 4.2.2 Use Case

Berdasarkan analisis kebutuhan sistem yang telah dilakukan, fungsionalitas sistem direpresentasikan kedalam beberapa *use case*. Daftar *use case* beserta pemetaannya sesuai kebutuhan fungsional dapat dilihat pada tabel 4.4.

Tabel 4.4 Pemetaan FR dan UC

FR ID	UC ID	Use Case
FR-03	UC-01	Login “Sales Obat”
FR-01	UC-02	Membuat pesanan obat baru
FR-01	UC-03	Menyimpan lokasi pemesan
FR-02	UC-04	Melihat daftar obat pesanan
FR-04	UC-05	Melihat daftar obat pesanan
FR-04	UC-06	Melihat lokasi pemesan
FR-05	UC-07	Mengupload foto dokumentasi
FR-03	UC-08	Logout “Sales Obat”
FR-06	UC-09	Login “Kurir Obat”
FR-06	UC-10	Logout “Kurir Obat”
FR-07	UC-11	Mengelola daftar pesanan
FR-08	UC-12	Menambah user aplikasi
FR-09	UC-13	Menentukan kurir pesanan
FR-10	UC-14	Melihat hasil <i>track</i> lokasi kurir
FR-11	UC-15	Melihat foto dokumentasi pengiriman pesanan
FR-12	UC-16	Login “Distributor Obat”
FR-12	UC-17	Logout “Distributor Obat”
FR-08	UC-18	Menghapus user aplikasi

*Use case diagram* dapat dilihat pada gambar 4.1.



Gambar 4.1 Use case diagram sistem

4.2.3 Use Case Description

UCD login “Sales Obat” dapat dilihat pada tabel 4.5

Tabel 4.5 UCD login “Sales Obat”

UC-01: Login “Sales Obat”	
Ringkasan:	Use case ini dimaksudkan agar sales dapat masuk kedalam aplikasi “Sales Obat” sesuai user yang dimiliki.
Aktor:	Sales

Kondisi:	Sales telah memiliki user yang dibuat oleh distributor
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Sales menjalankan aplikasi</li> <li>2. Sales mengetikkan user dan password</li> <li>3. Sales menekan tombol login</li> <li>4. Sales berhasil masuk ke halaman utama aplikasi</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Sales mengetikkan user atau password yang tidak sesuai atau belum terdaftar, maka sistem akan menampilkan pesan eror.</li> </ol>

UCD membuat pesanan obat baru dapat dilihat pada tabel 4.6.

Tabel 4.6 UCD membuat pesanan obat baru

UC-02: Membuat pesanan obat baru	
Ringkasan:	Use case ini dimaksudkan agar sales dapat membuat pesanan obat baru.
Aktor:	Sales
Kondisi:	Sales telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Sales memilih menu “tambah pesanan”</li> <li>2. Sales memasukkan nama obat beserta jumlahnya.</li> <li>3. Sales menekan tombol “next” maka aplikasi akan menampilkan halaman pengisian informasi pemesanan</li> </ol>

UCD menyimpan lokasi pemesan dapat dilihat pada tabel 4.7.

Tabel 4.7 UCD menyimpan lokasi pemesan

UC-03: Menyimpan lokasi pemesan
---------------------------------



Ringkasan:	Use case ini dimaksudkan bahwa sales akan menyimpan lokasi pemesan pada saat membuat pesanan baru
Aktor:	Sales
Kondisi:	Sales telah melakukan UC-02
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Sales memasukkan nama apotek atau rumah sakit pemesan</li> <li>2. Sales menekan lokasi pemesan pada google map.</li> <li>3. Sales melengkapi alamat pemesan</li> <li>4. Sales menekan tombol “simpan”, maka aplikasi akan menyimpan pesanan baru tersebut.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Sales tidak memilih lokasi pemesan pada google maps, maka aplikasi akan menampilkan pesan eror pada saat melakukan penyimpanan pesanan.</li> </ol>

UCD melihat daftar obat pesanan oleh sales dapat dilihat pada tabel 4.8.

Tabel 4.8 UCD melihat daftar obat pesanan

UC-04: Melihat daftar obat pesanan	
Ringkasan:	Use case ini dimaksudkan bahwa sales dapat melihat daftar obat dari suatu pesanan yang telah dibuat sebelumnya.
Aktor:	Sales
Kondisi:	Sales telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Sales memilih salah satu pesanan dari daftar pesanan</li> <li>2. Sales dapat melihat daftar obat pada pesanan tersebut</li> </ol>

UCD melihat daftar obat pesanan oleh kurir dapat dilihat pada tabel 4.9.

Tabel 4.9 UCD melihat daftar obat pesanan

UC-05: Melihat daftar obat pesanan	
Ringkasan:	Use case ini dimaksudkan bahwa kurir dapat melihat daftar obat dari suatu pesanan
Aktor:	Kurir
Kondisi:	Kurir telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Kurir memilih salah satu pesanan dari daftar</li> <li>2. Kurir dapat melihat daftar obat pada pesanan tersebut</li> </ol>

UCD melihat lokasi pemesan dapat dilihat pada tabel 4.10.

Tabel 4.10 UCD melihat lokasi pemesan

UC-06: Melihat lokasi pemesan	
Ringkasan:	Use case ini dimaksudkan bahwa kurir dapat melihat lokasi pemesan dalam google maps.
Aktor:	Kurir
Kondisi:	Kurir telah melakukan UC-05
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Kurir menekan tombol “Lokasi”</li> <li>2. Kurir dapat melihat lokasi pemesan dalam google maps.</li> </ol>

UCD mengupload foto dokumentasi dapat dilihat pada tabel 4.11.

Tabel 4.11 UCD mengupload foto dokumentasi

UC-07: Mengupload foto dokumentasi
------------------------------------

Ringkasan:	Use case ini dimaksudkan bahwa kurir dapat mengupload foto dokumentasi pada saat menyerahkan obat pesanan.
Aktor:	Kurir
Kondisi:	Kurir telah melakukan UC-05
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Kurir menekan tombol “Upload foto”</li> <li>2. Kurir menekan tombol “Ambil foto”</li> <li>3. Kurir menekan tombol “Upload foto” setelah foto diambil, maka aplikasi akan menyimpan foto tersebut serta mengubah status pesanan menjadi “selesai”</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Jika terjadi kesalahan koneksi pada saat mengupload foto, maka aplikasi akan menampilkan pesan eror.</li> </ol>

UCD logout “Sales Obat” dapat dilihat pada tabel 4.12.

Tabel 4.12 UCD logout “Sales Obat”

UC-08: Logout “Sales Obat”	
Ringkasan:	Use case ini dimaksudkan agar sales dapat keluar dari aplikasi “Sales Obat”.
Aktor:	Sales
Kondisi:	Sales telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Sales memilih menu “logout”</li> <li>2. Sales keluar dari aplikasi</li> </ol>

UCD login “Kurir Obat” dapat dilihat pada tabel 4.13.

Tabel 4.13 UCD login “Kurir Obat”

UC-09: Login “Kurir Obat”
---------------------------

Ringkasan:	Use case ini dimaksudkan agar kurir dapat masuk kedalam aplikasi “Kurir Obat” sesuai user yang dimiliki.
Aktor:	Kurir
Kondisi:	Kurir telah memiliki user yang dibuat oleh distributor
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Kurir menjalankan aplikasi</li> <li>2. Kurir mengetikkan user dan password</li> <li>3. Kurir menekan tombol login</li> <li>4. Kurir berhasil masuk ke halaman utama aplikasi</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Kurir mengetikkan user atau password yang tidak sesuai atau belum terdaftar, maka sistem akan menampilkan pesan error.</li> </ol>

UCD logout “Kurir Obat” dapat dilihat pada tabel 4.14.

Tabel 4.14 UCD logout “Kurir Obat”

UC-10: Logout “Kurir Obat”	
Ringkasan:	Use case ini dimaksudkan agar kurir dapat keluar dari aplikasi “Kurir Obat”.
Aktor:	Kurir
Kondisi:	Kurir telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Kurir memilih menu “logout”</li> <li>2. Kurir telah keluar dari aplikasi</li> </ol>

UCD mengelola daftar pesanan dapat dilihat pada tabel 4.15

Tabel 4.15 UCD mengelola daftar pesanan

UC-11: Mengelola daftar pesanan	
Ringkasan:	Use case ini dimaksudkan bahwa distributor dapat mengelola daftar pesanan yang ada.

Aktor:	Distributor
Kondisi:	Distributor telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Distributor memilih menu daftar pesanan</li> <li>2. Distributor menekan tombol “Hapus” pada suatu pesanan.</li> <li>3. Distributor menekan tombol “YA” pada tampilan konfirmasi penghapusan, maka pesanan telah dihapus.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Distributor menekan tombol “Tidak” pada tampilan konfirmasi penghapusan, maka pesanan tidak akan dihapus.</li> </ol>

UCD menambah user aplikasi dapat dilihat pada tabel 4.16.

Tabel 4.16 UCD menambah user aplikasi

UC-12: Menambah user aplikasi	
Ringkasan:	Use case ini dimaksudkan bahwa distributor dapat menambah user dari setiap aplikasi
Aktor:	Distributor
Kondisi:	Distributor telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Distributor memilih menu “Kelola User”</li> <li>2. Distributor menekan tombol “tambah user”</li> <li>3. Distributor mengisi user dan password yang akan digunakan</li> <li>4. Distributor menekan tombol simpan user, maka user tersebut akan disimpan</li> </ol>

UCD menentukan kurir pesanan dapat dilihat pada tabel 4.17.

Tabel 4.17 UCD menentukan kurir pesanan

UC-13: Login “Sales Obat”	
Ringkasan:	Use case ini dimaksudkan bahwa distributor dapat menentukan kurir yang bertugas untuk mengirim pesanan.
Aktor:	Distributor
Kondisi:	Distributor telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Distributor memilih menu “daftar pesanan”</li> <li>2. Distributor menekan tombol “<i>assign</i> kurir”</li> <li>3. Distributor memilih user kurir yang akan mengirim pesanan.</li> <li>4. Distributor menekan tombol “Simpan”, maka aplikasi akan mengirim notifikasi pesanan kepada kurir.</li> </ol>

UCD melihat hasil *track* lokasi kurir dapat dilihat pada tabel 4.18.

Tabel 4.18 UCD melihat hasil *track* lokasi kurir

UC-14: Melihat hasil <i>track</i> lokasi kurir	
Ringkasan:	Use case ini dimaksudkan bahwa distributor dapat melihat hasil <i>track</i> lokasi kurir pada google maps dari aplikasi.
Aktor:	Distributor
Kondisi:	Distributor telah login kedalam aplikasi

Skenario Utama:	<ol style="list-style-type: none"> <li>1. Distributor memilih menu “<i>Track kurir</i>”</li> <li>2. Distributor memilih user kurir yang akan dilihat.</li> <li>3. Distributor memilih tanggal <i>track</i> lokasi kurir.</li> <li>4. Distributor dapat melihat hasil <i>track</i> lokasi kurir pada google maps didalam aplikasi.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Distributor memilih tanggal <i>track</i> yang tidak memiliki data, maka aplikasi akan menampilkan pesan eror.</li> </ol>

UCD melihat foto dokumentasi pengiriman pesanan dapat dilihat pada tabel 4.19.

Tabel 4.19 UCD melihat foto dokumentasi pengiriman pesanan

UC-15: Melihat foto dokumentasi pengiriman pesanan	
Ringkasan:	Use case ini dimaksudkan bahwa distributor dapat melihat foto dokumentasi pengiriman pesanan.
Aktor:	Distributor
Kondisi:	Distributor telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Distributor memilih menu “daftar pesanan selesai”</li> <li>2. Distributor menekan tombol “Foto dokumentasi” pada salah satu pesanan</li> <li>3. Distributor dapat melihat foto dokumentasi penyerahan obat pesanan.</li> </ol>

UCD login “Distributor Obat dapat dilihat pada tabel 4.20.

Tabel 4.20 UCD login “Distributor Obat”

UC-16: Login “Distributor Obat”	
Ringkasan:	Use case ini dimaksudkan agar distributor dapat masuk kedalam aplikasi “Distributor Obat” sesuai user yang dimiliki.
Aktor:	Distributor
Kondisi:	Distributor telah memiliki user dan password
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Distributor menjalankan aplikasi</li> <li>2. Distributor mengetikkan user dan password</li> <li>3. Distributor menekan tombol login</li> <li>4. Distributor berhasil masuk ke halaman utama aplikasi</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Sales mengetikkan user atau password yang tidak sesuai, maka sistem akan menampilkan pesan eror.</li> </ol>

UCD logout “Distributor Obat” dapat dilihat pada tabel 4.21.

Tabel 4.21 UCD login “Sales Obat”

UC-17: Logout “Distributor Obat”	
Ringkasan:	Use case ini dimaksudkan agar distributor dapat keluar dari aplikasi “Distributor Obat”.
Aktor:	Distributor
Kondisi:	Distributor telah melakukan login
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Distributor memilih menu “Logout”</li> <li>2. Distributor telah keluar dari aplikasi.</li> </ol>

UCD menghapus user aplikasi dapat dilihat pada tabel 4.22.



Tabel 4.22 UCD menghapus user aplikasi

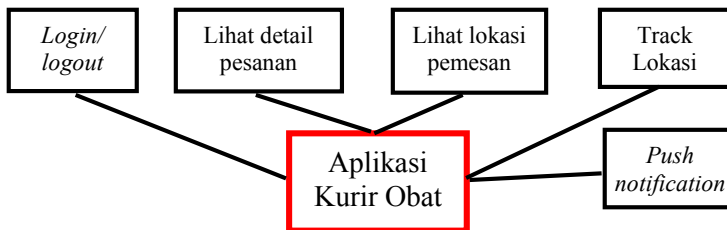
UC-18: Menghapus user aplikasi	
Ringkasan:	Use case ini dimaksudkan bahwa distributor dapat menghapus user dari sales maupun kurir
Aktor:	Distributor
Kondisi:	Distributor telah login kedalam aplikasi
Skenario Utama:	<ol style="list-style-type: none"> <li>1. Distributor memilih menu “Kelola user”</li> <li>2. Distributor menekan tombol “Hapus” pada salah satu user.</li> <li>3. Distributor menekan tombol “YA” pada tampilan konfirmasi penghapusan user, maka user tersebut telah dihapus.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Distributor menekan tombol “Tidak” pada tampilan konfirmasi penghapusan user, maka user tersebut tidak dihapus.</li> </ol>

#### 4.2.4 Pembagian Fungsi Aplikasi

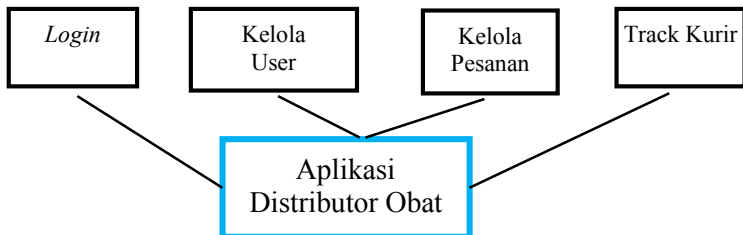
Setiap fungsi aplikasi akan dibagi sesuai dengan pembagian aktor yang menggunakan. Pembagian fungsi pada aplikasi sales obat dapat dilihat pada gambar 4.2, aplikasi kurir obat pada gambar 4.3, dan aplikasi distributor obat pada gambar 4.4.



Gambar 4.2 Fungsi pada aplikasi sales obat



Gambar 4.3 Fungsi pada aplikasi kurir obat



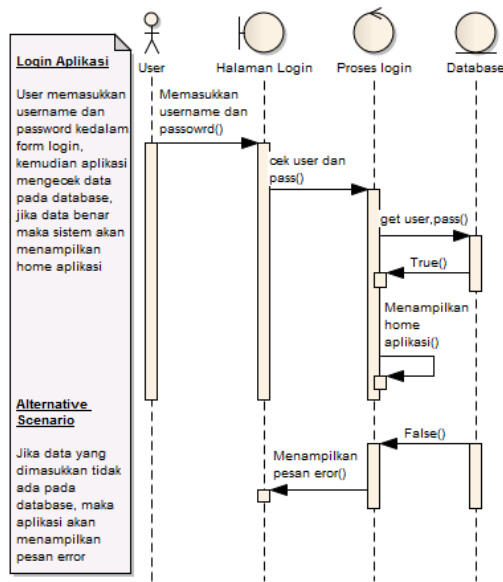
Gambar 4.4 Fungsi pada aplikasi distributor obat

#### 4.2.5 Sequence Diagram

*Sequence diagram* dari aplikasi monitoring kurir ini secara keseluruhan meliputi alur-alur *method* dan *object* dari setiap aplikasi bagian Sales Obat, Kurir Obat, dan Distributor Obat yang dijabarkan sebagai berikut.

1. *Login* aplikasi

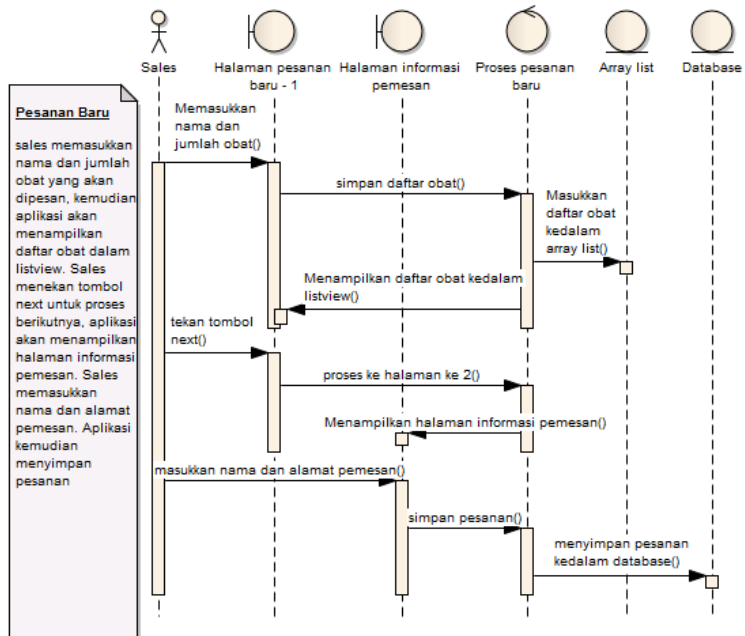
Setiap user yang akan menggunakan aplikasi terlebih dahulu harus melakukan *login*, proses tersebut dapat dilihat pada gambar 4.5.



Gambar 4.5 Sequence diagram login

## 2. Buat pesanan baru

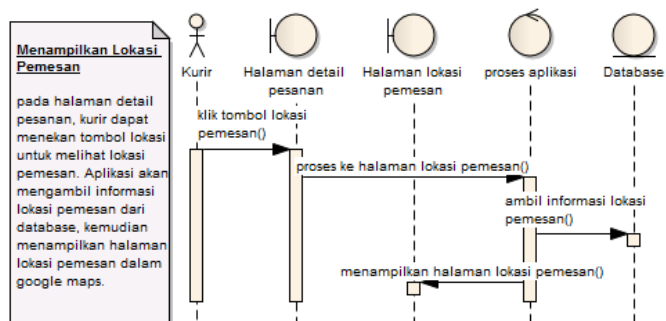
Proses pembuatan pesanan baru oleh sales dapat dilihat pada gambar 4.6. Aplikasi akan menampilkan *form* pesanan, sales memasukkan nama dan jumlah obat yang akan dipesan kemudian aplikasi akan menampilkan daftar obat dalam *listview*. Sales menekan tombol next untuk proses berikutnya, aplikasi akan menampilkan halaman informasi pemesan. Sales memasukkan nama dan alamat pemesan. Aplikasi kemudian menyimpan pesanan kedalam database.



Gambar 4.6 Sequence diagram pesanan baru

### 3. Lokasi pemesanan

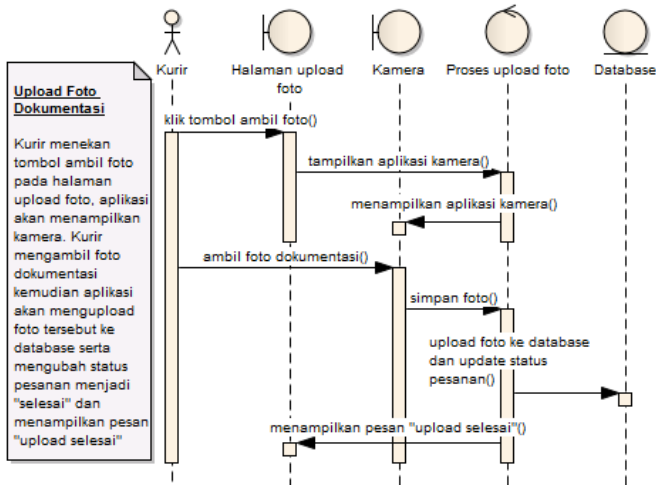
Aplikasi dapat menampilkan lokasi pemesanan dengan proses yang dapat dilihat pada gambar 4.7.



Gambar 4.7 Sequence diagram lokasi pemesanan

#### 4. *Upload* foto dokumentasi

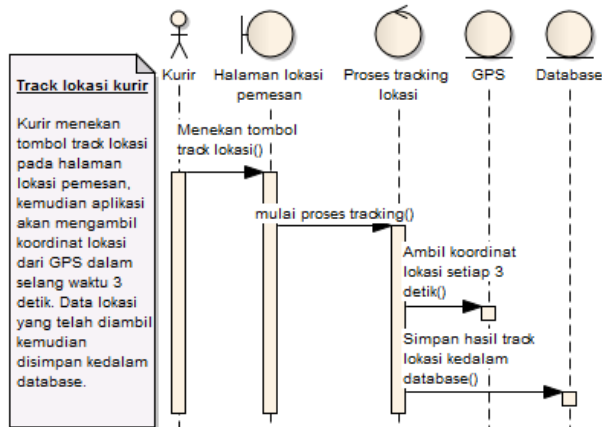
Kurir yang sudah mengirim pesanan dapat meng-*upload* foto dokumentasi dengan proses pada gambar 4.8.



Gambar 4.8 *Sequence diagram upload* foto dokumentasi

#### 5. *Track* lokasi kurir

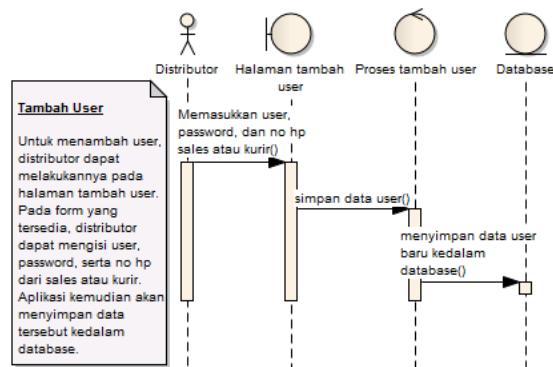
Kurir yang sedang mengirim pesanan dapat di-*track* lokasinya dengan memanfaatkan teknologi GPS. Aplikasi akan menyimpan lokasi kurir dalam selang waktu 3 dan kemudian menyimpannya kedalam database. Proses *tracking* lokasi kurir dapat dilihat pada gambar 4.9.



Gambar 4.9 *Sequence diagram track lokasi kurir*

## 6. Menambah user

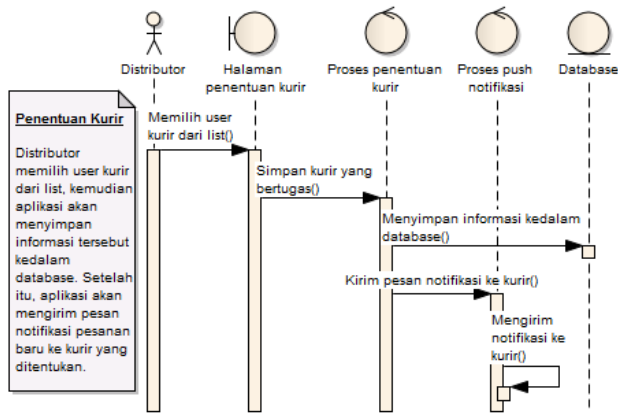
Distributor dapat menambah *user* yang akan digunakan oleh sales maupun kurir. Proses ini dapat dilihat pada gambar 4.10.



Gambar 4.10 *Sequence diagram menambah user*

## 7. Menentukan kurir yang bertugas

Kurir yang bertugas untuk mengirim pesanan baru dapat ditentukan oleh distributor dengan proses pada gambar 4.11.



Gambar 4.11 Sequence diagram penentuan kurir

#### 4.2.6 Daftar Class dan Method

Setiap nama *class* dan *method* yang akan digunakan pada pembuatan aplikasi akan dijabarkan di sub bab ini. Penjabaran akan dibagi menjadi tiga, yakni untuk aplikasi Sales Obat, aplikasi Kurir Obat, dan aplikasi Distributor Obat.

##### 1. Class dan method sales obat

*Class* dan *method* yang digunakan pada pembuatan aplikasi sales obat dijabarkan pada tabel 4.23.

Tabel 4.23 Daftar nama *class* dan *method* aplikasi sales obat

Class	Method
Login	onCreate(); login(); getMD5();
Home	onCreate(); logout(); menu(); viewPager();

prosesFragment	onCreate(); fetchPesanan();
selesaiFragment	onCreate(); fetchPesanan();
tambahPesanan	onCreate(); tambahPesanan(); fetchlist(); optionmenu();
tambahPesananCont	onCreate(); checkGPS(); pesananbaru(); onMapClick(); onMapReady(); optionmenu(); getAddressfromLatLng();

## 2. *Class* dan *method* kurir obat

*Class* dan *method* yang digunakan pada pembuatan aplikasi kurir obat dijabarkan pada tabel 4.24.

Tabel 4.24 Daftar nama *class* dan *method* aplikasi kurir obat

<b>Class</b>	<b>Method</b>
Login	onCreate(); getMD5Hash(); login(); onClick();
Home	onCreate(); logout(); update_gcm(); setUpViewPager();
prosesFragment	onCreateView(); syncLocation(); fetchPesanan(); onRefresh();



selesaiProses	onCreateView(); fetchPesanan(); onRefresh();
Lokasi	onCreate(); fetchLokasi(); onMapReady(); onConnected(); initCamera(); checkGPS(); getLocation(); route(); getAddressFromLatLng();

### 3. Distributor obat

Pada aplikasi distributor obat, pengembangan dilakukan dengan orientasi proses. Aplikasi dikembangkan dengan tujuan menghasilkan fitur yang dapat digunakan sesuai dengan rancangan yang dijelaskan sebelumnya.

#### **4.2.7 Desain Interface**

Desain *Interface* aplikasi diperlukan untuk menggambarkan sketsa aplikasi yang akan dirancang dengan tujuan kemudahan dalam menentukan alur kerja sistem. Desain *interface* menggunakan aplikasi pengolah foto Adobe Photoshop dan tambahan template android. Berikut adalah beberapa desain interface yang akan digunakan dalam pembuatan aplikasi ini:

##### **4.2.7.1. Aplikasi Sales Obat**

###### 1. *Login*

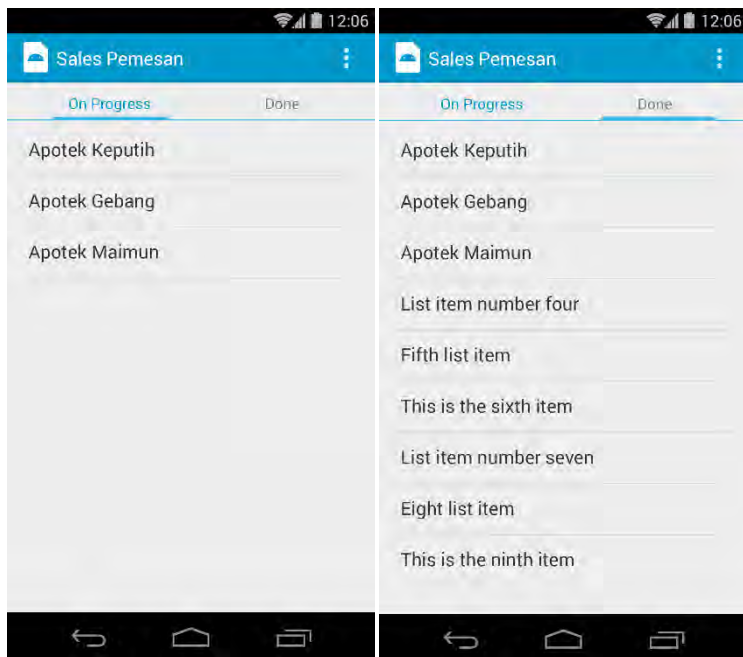
Aplikasi sales obat ketika pertama kali dijalankan akan menampilkan halaman *login*. Sales akan diminta untuk memasukkan *username* dan *password* yang didapat dari distributor. Desain halaman *login* dapat dilihat pada gambar 4.12 berikut ini.



Gambar 4.12 Halaman *login* aplikasi sales obat

## 2. Home

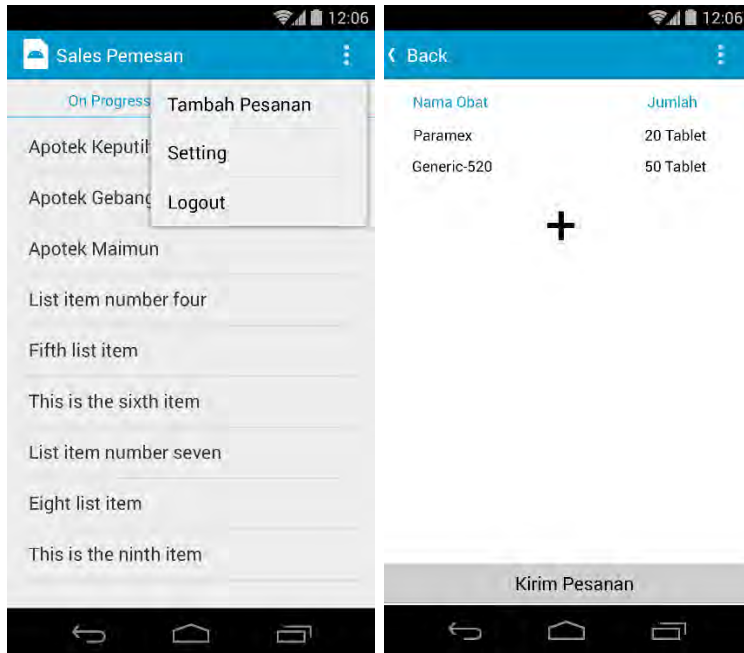
Setelah sales berhasil melakukan *login*, aplikasi akan menampilkan *tab on-Progress*. Halaman ini berisi pesanan yang dibuat oleh sales tersebut dan dalam status pengiriman. Pada halaman *home* ini juga terdapat *tab Done* yang dapat diakses dengan menggeser halaman sebelah kanan. *Tab Done* akan menampilkan daftar pesanan yang dibuat oleh sales tersebut dan dalam status sudah dikirim. Desain *tab on-progress* dan *Done* dapat dilihat pada gambar 4.13



Gambar 4.13 *Tab On Progress* dan *Done* aplikasi

### 3. Pesanan Baru

Pesanan baru dapat dibuat dengan memilih menu tambah pesanan. Pada halaman tambah pesanan, sales dapat mengisi informasi pesanan berupa nama obat beserta jumlahnya. Desain menu dan halaman tambah pesanan dapat dilihat pada gambar 4.14.



Gambar 4.14 Menu aplikasi dan Halaman Tambah Pesanan

#### 4.2.4.2. Aplikasi Kurir Obat

Desain interface aplikasi kurir obat tidak jauh berbeda dengan aplikasi sales obat. Ketika aplikasi pertama sekali dijalankan, maka akan menampilkan halaman login. Setelah kurir berhasil melakukan login, aplikasi akan menampilkan daftar pesanan yang harus dikirim dalam tab on-progress dan pesanan yang sudah dikirim dalam tab Done.

##### 1. Informasi pesanan

Kurir dapat mengakses informasi pesanan dengan memilih salah satu pesanan dari daftar, aplikasi akan menampilkan informasi pesanan seperti daftar obat pesanan beserta

jumlahnya. Desain halaman informasi pesanan dapat dilihat pada gambar 4.15 berikut ini.



Gambar 4.15 Desain halaman informasi pesanan

## 2. Foto bukti

Setelah mengirim pesan ke apotek atau rumah sakit pemesan, kurir dapat mengambil foto dan mengupload ke server dengan mengakses halaman foto bukti. Fitur ini dapat menjadi kemudahan dalam dokumentasi setiap pengiriman pesanan. Desain halaman foto bukti dapat dilihat pada gambar 4.16.



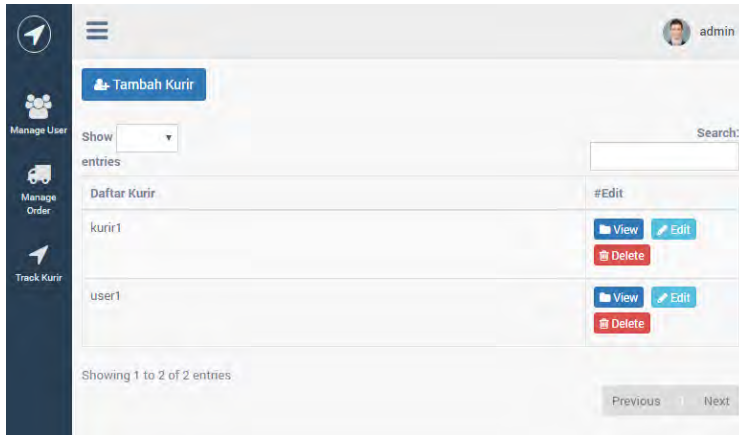
Gambar 4.16 Halaman foto bukti

Halaman lokasi ketika diakses akan menampilkan google maps beserta dengan lokasi pemesan.

#### 4.2.4.3. Aplikasi Distributor Obat

##### 1. Kelola user

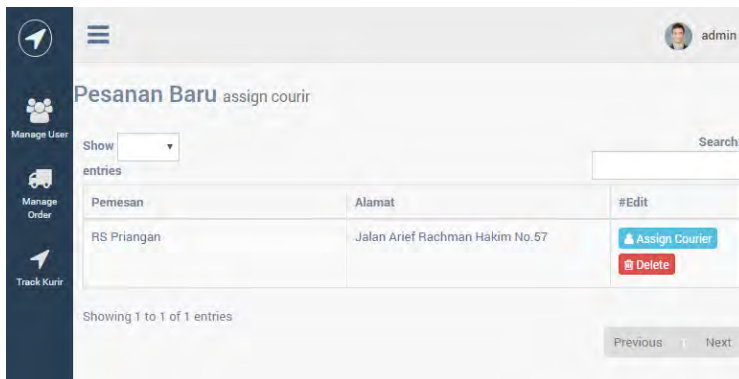
Aplikasi web memiliki fungsi untuk mengelola user dari setiap aplikasi android yakni untuk sales dan kurir. Pada halaman pengelolaan user ini, ditampilkan daftar user yang dibagi menjadi user kurir dan user sales. Desain dari halaman pengelolaan user ini dapat dilihat pada gambar 4.17.



Gambar 4.17 Halaman pengelolaan user

## 2. Kelola pesanan

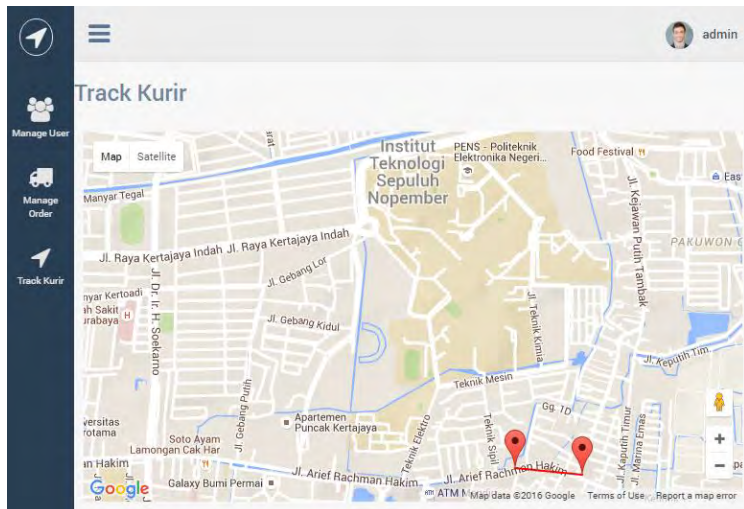
Pengelolaan pesanan dibagi menjadi tiga bagian yakni penentuan kurir, pesanan dalam proses pengiriman, dan pesanan yang sudah dikirim. Ketiga bagian tersebut ditampilkan dalam halaman yang berbeda-beda. Desain pengelolaan pesanan dapat dilihat pada gambar 4.18.



Gambar 4.18 Halaman pengelolaan pesanan

### 3. *Track* kurir

Data lokasi yang direkam oleh aplikasi kurir ditampilkan pada halaman track kurir. Pilih kurir yang akan dilihat data lokasinya terlebih dahulu, kemudian pilih tanggal data lokasi. Pergerakan lokasi kurir akan ditampilkan dalam bentuk garis dalam google maps. Desain halaman track kurir dapat dilihat pada gambar 4.19 berikut ini.



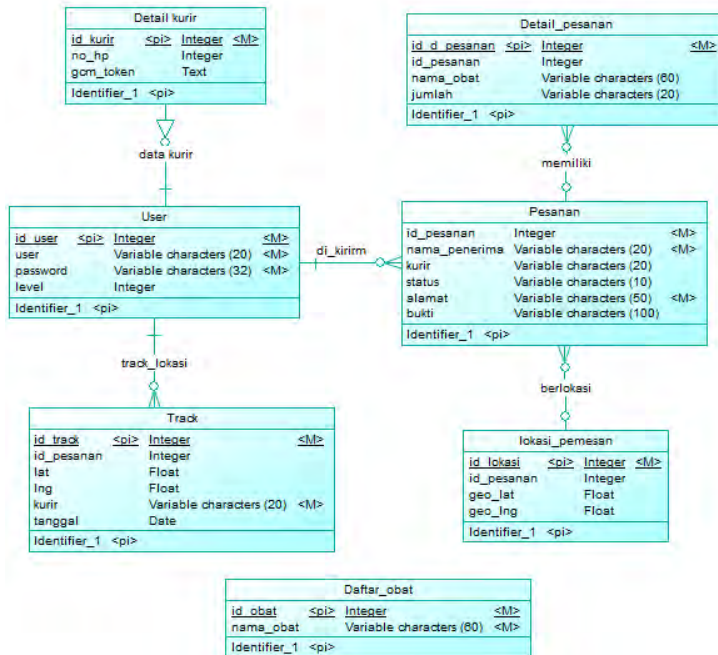
Gambar 4.19 Halaman Track Kurir

## 4.3 Desain Database

Database untuk aplikasi monitoring kurir secara keseluruhan dibuat dengan pemodelan *Conceptual Data Model* (CDM) dan kemudian dihasilkan *Physical Data Model* (PDM). Pembuatan CDM didasarkan pada kebutuhan sistem yang sudah dijabarkan sebelumnya. CDM dari aplikasi monitoring kurir dapat dilihat pada gambar 4.20. CDM dibuat dengan menggunakan *software* Sybase Power Designer versi 16.1.



Pada gambar CDM tersebut, terdapat satu entitas daftar obat yang tidak memiliki relasi dengan entitas lainnya. Entitas ini merupakan tabel yang akan menampung daftar nama-nama obat yang dimiliki oleh distributor. Fitur pencarian nama obat pada aplikasi pemesanan obat untuk sales akan mengambil data dari tabel daftar obat ini.



Gambar 4.20 Conceptual Data Model aplikasi

Penjelasan mengenai entitas pada gambar 4.20 dijabarkan pada tabel 4.25 berikut ini.

Tabel 4.25 Penjelasan Entitas pada CDM

No	Entitas	Atribut	Keterangan
1	User	id_user user	id user sebagai PK pada entitas ini,

		password level	user dan password digunakan untuk <i>login</i> , level digunakan untuk membedakan status user.
2	Detail Kurir	id_kurir no_hp gcm_token	id_kurir sebagai PK entitas. no_hp untuk menyimpan no HP kurir, gcm_token untuk menyimpan kode registrasi <i>push notification</i>
3	Track Posisi	id_track id_pesanan lat lng kurir tanggal	id_track sebagai PK id_pesanan sebagai FK untuk entitas Pesanan. lat & lng adalah latitude dan longitude untuk <i>track</i> lokasi kurir. kurir sebagai pemilik data. tanggal untuk membedakan setiap rekaman lokasi.
4	Detail Pesanan	id_d_pesanan id_pesanan nama_obat jumlah	id_d_pesanan, sebagai PK. id_pesanan, sebagai FK untuk entitas pesanan. nama_obat, untuk menyimpan nama obat pesanan. jumlah, untuk menyimpan jumlah obat pesanan.
5	Pesanan	id_pesanan nama_penerima kurir	id_pesanan, sebagai PK. nama_penerima, untuk menyimpan nama

		status alamat bukti	instansi pemesan. kurir, menyimpan user kurir yang bertugas. status, menyimpan status pesanan. alamat, menyimpan alamat instansi pemesan. bukti, menyimpan <i>path</i> file foto dokumentasi.
6	Lokasi Pemesan	id_lokasi id_pesanan geo_lat geo_lng	id_lokasi, sebagai PK. id_pesanan, sebagai FK untuk entitas pesanan. geo_lat & geo_lng, untuk menyimpan lokasi instansi pemesan.
7	Daftar Obat	id_obat nama_obat	id_obat, sebagai PK. nama_obat, untuk menyimpan nama obat yang dijual oleh distributor.

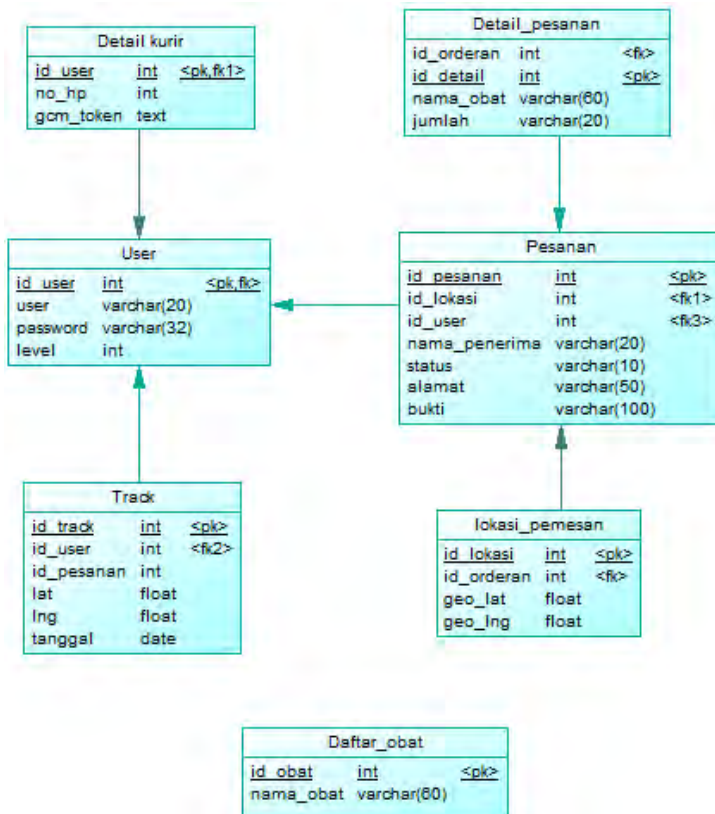
Relasi antar entitas pada CDM dijabarkan pada tabel 4.26 berikut ini.

Tabel 4.26 Relasi antar Entitas CDM

Relationship Type	Constraint	Keterangan
DATA_KURIR on DETAIL_KURIR, USER	(1,1)	user bisa memiliki maksimal satu detail kurir dan satu detail kurir harus memiliki maksimal satu user.
TRACK_LOKASI on USER, TRACK	(1,N)	Setiap user bisa memiliki banyak track posisi dan Setiap track harus

		memiliki maksimal satu user.
DIKIRIM on PESANAN, USER	(N,1)	Setiap pesanan harus memiliki maksimal satu user dan satu user bisa mengantar banyak pesanan.
MEMILIKI on PESANAN, DETAIL_PESANAN	(1,N)	Setiap pesanan dapat memiliki banyak detail pesanan dan setiap detail pesanan harus memiliki maksimal satu pesanan
BERLOKASI on PESANAN, LOKASI_PEMESAN	(N,1)	Beberapa pesanan dapat berlokasi dalam satu lokasi pemesan dan satu lokasi dapat menjadi lokasi dari beberapa pesanan.

*Physical Data Model* yang di-generate dari CDM tersebut dapat dilihat pada gambar 4.21.



Gambar 4.21 *Physical Data Model* aplikasi

PDM yang dihasilkan tidak memiliki perbedaan tabel dari CDM, karena setiap entitas yang dibutuhkan sudah terpenuhi dengan baik. PDM dapat di-*generate* menjadi file sql yang siap untuk di-*import* pada database Mysql.

### 4.3.1 Pemetaan Penggunaan Tabel Database

Setiap aplikasi menggunakan tabel sesuai dengan kebutuhan masing-masing yang dijabarkan sebagai berikut.

### 1. Aplikasi Sales Obat

Aplikasi sales obat menggunakan beberapa tabel untuk menyimpan data, yakni tabel user, daftar obat, pesanan, detail pesanan, dan lokasi pemesan.

### 2. Aplikasi Kurir Obat

Aplikasi kurir obat menggunakan beberapa tabel untuk menyimpan data, yakni tabel user, detail kurir, pesanan, detail pesanan, lokasi pemesan, dan track.

### 3. Aplikasi Distributor Obat

Aplikasi distributor obat menggunakan beberapa tabel untuk menyimpan data, yakni tabel user, pesanan, detail pesanan, dan track.

*Halaman ini sengaja dikosongkan*

## **BAB V**

### **IMPLEMENTASI**

Pada bab ini akan dijelaskan mengenai implementasi dari perancangan yang telah dilakukan sesuai dengan metode pengembangan yang dibuat. Bagian implementasi akan menjelaskan mengenai pembuatan daftar obat, lingkungan implementasi aplikasi, pembuatan fitur-fitur aplikasi dalam bentuk kode, serta pengujian aplikasi.

#### **5.1 Daftar Obat**

Pembuatan daftar obat bertujuan untuk melengkapi fitur pencarian nama obat, sehingga mempermudah sales dalam memasukkan obat pesanan. Aplikasi ini juga dapat digunakan untuk jenis produk lainnya, seperti makanan atau minuman. Namun untuk konteks dalam tugas akhir ini yakni distribusi obat, maka daftar produk yang dimasukkan adalah jenis-jenis obat yang dijual oleh distributor.

Daftar obat yang akan dimuat dalam *database* aplikasi ini merupakan contoh implemetasi, dan menggunakan nama sesuai dengan referensi. Daftar nama obat diambil dari website penyedia layanan kesehatan Kalbe Farma [13]. Nama-nama obat yang akan dimasukkan pada *database* merupakan jenis obat *generic*.

#### **5.2 Lingkungan Implementasi**

Pengembangan aplikasi ini menggunakan komputer dengan spesifikasi pada tabel 5.1 berikut ini.

Tabel 5.1 Spesifikasi Komputer

<b>Prosesor</b>	Intel® Core™ i3-2348M @ 2.30GHz
-----------------	---------------------------------



<b>Memory</b>	8000 MB RAM
<b>Sistem Operasi</b>	Windows 8.1 (64-bit)

Aplikasi dikembangkan dengan menggunakan beberapa teknologi seperti *Software Development Kit*, *editor*, *database*, *server*, bahasa pemrograman dan *library* yang disajikan dalam tabel 5.2 berikut.

Tabel 5.2 Teknologi yang Digunakan

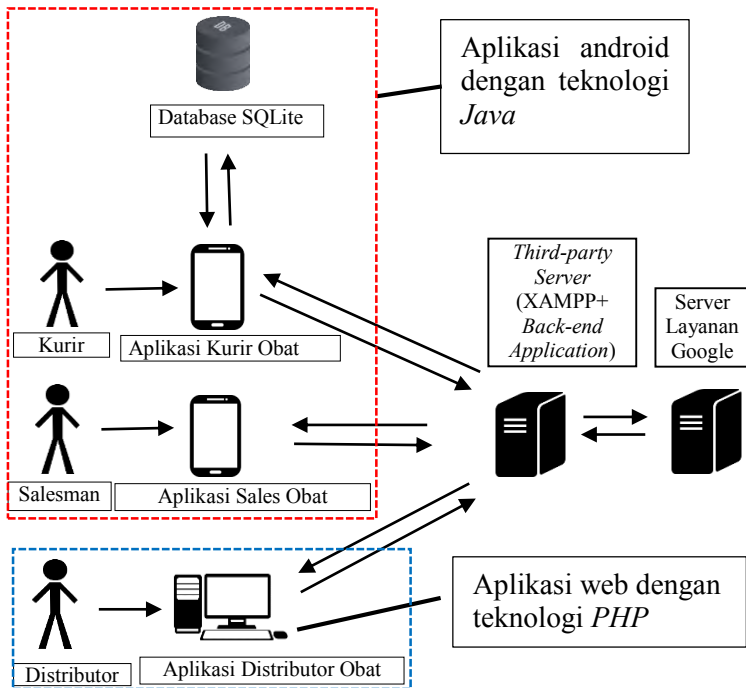
<b>Webserver</b>	Apache 2.4
<b>Bahasa Pemrograman</b>	PHP 5.6.3, Java, Javascript
<b>Database</b>	MySql 5.6.21
<b>SDK</b>	Android SDK 1.5.1
<b>Editor</b>	Notepad ++ 6.7.9.2
<b>Browser</b>	Google Chrome 51.0
<b>Library</b>	-jQuery v2.2.3 (Javascript Lib) -Bootstrap v3.3.6 (Layout) -FullCalendar v2.7.3 (Calendar) -FontAwesome v4.6.3 (Icon) -DataTables v1.10.11 (Tabel)

Untuk melakukan uji coba aplikasi android, maka digunakan *mobile phone* dengan spesifikasi pada tabel 5.3 berikut.

Tabel 5.3 Spesifikasi *Mobile Phone*

Sistem Operasi	Android Lollipop 5.0.2
CPU	Octacore 2.0GHz
RAM	2000 MB
Internal Memory	16 GB
GPS	Yes
Network Connection	Yes
Camera	Yes

Pengembangan aplikasi sesuai dengan teknologi yang digunakan dapat dilihat pada gambar 5.1 berikut.



Gambar 5.1 Arsitektur pengembangan aplikasi

## 5.3 Pembuatan Aplikasi Sales Obat

Pada bagian ini akan dijelaskan pembuatan aplikasi android sales obat dalam bentuk kode program yang digunakan sesuai dengan desain aplikasi.

### 5.3.1 Fungsi Login

Sales akan menggunakan kombinasi *username* dan *password* untuk melakukan *login* ke aplikasi. *Login* dalam aplikasi pemesanan obat menggunakan *library volley* untuk

menjalankan file *login.php* yang ada pada server. Halaman *login* akan muncul ketika pertama kali menjalankan aplikasi. Tampilan halaman login ada pada gambar 5.2.



Gambar 5.2 Halaman *login* aplikasi sales obat

File *login.php* akan dipanggil berisi fungsi untuk melakukan pengecekan *username* dan *password* yang ada pada *database*. Untuk meningkatkan keamanan *query* yang dijalankan, maka digunakan *sql prepared statement* agar *query* yang dijalankan tidak terpengaruh dengan *input* yang dimasukkan. Penggunaan *sql prepared statement* dapat dilihat pada gambar 5.3.

```
$query = $con->prepare("SELECT * FROM user WHERE
BINARY user=? AND BINARY password=?");

$query->bind_param('ss',$user,$password);

$query->execute();

$query->bind_result($id,$use,$passe,$level);
```

Gambar 5.3 Penggunaan *sql prepared statement*

Jika sales berhasil *login* ke aplikasi, maka *username* sales tersebut akan disimpan dalam satu *class* dengan menggunakan *Sharedpreferences*. *Class* yang menyimpan *username* tersebut

akan digunakan sebagai penanda bahwa *user* tersebut telah login. Penggunaan *SharedPreferences* untuk menyimpan *username* sales dalam suatu *class* dapat dilihat pada gambar 5.4.

```
SharedPreferences sharedPreferences = MainActivity.this.getSharedPreferences
    (Config.SHARED_PREF_NAME, Context.MODE_PRIVATE);

//Creating editor to store values to shared preferences
SharedPreferences.Editor editor = sharedPreferences.edit();

//Adding values to editor
editor.putBoolean(Config.LOGGEDIN_SHARED_PREF, true);
editor.putString(Config.USER_SHARED_PREF, user);

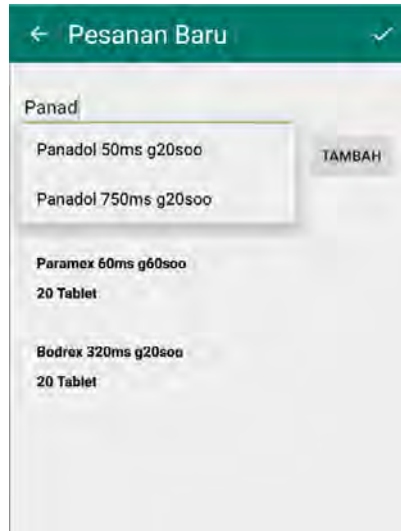
//Saving values to editor
editor.apply();
```

Gambar 5.4 Penggunaan *SharedPreferences* pada aplikasi

Setelah menyimpan *username* tersebut, aplikasi akan melakukan *intent* untuk menjalankan *class* yang akan menampilkan *home* aplikasi.

### 5.3.2 Fungsi Buat Pesanan Baru

Untuk membuat pesanan baru, dibutuhkan *class* yang akan menampilkan halaman pengisian nama obat beserta beserta jumlahnya. Ketika memasukkan nama obat yang akan dimasukkan ke dalam daftar pesanan, aplikasi akan menampilkan hasil pencarian nama obat dari *database* server. Proses pencarian akan dimulai setelah *input* nama obat minimal empat karakter. Hal tersebut dilakukan untuk meminimalisir *resource* yang dibutuhkan untuk menampilkan hasil pencarian. Tampilan halaman buat pesanan baru dapat dilihat pada gambar 5.5.



Gambar 5.5 Pesanan Baru

Setiap obat yang ditambahkan akan disimpan kedalam *arraylist* kemudian ditampilkan oleh *listview*. Penggunaan metode seperti ini membuat daftar pesanan bersifat fleksibel, sehingga dapat menampilkan daftar pesanan obat dalam jumlah banyak atau sedikit. Proses penyimpanan obat pesanan dimulai setelah menekan tombol “tambah”. Kode program untuk memasukkan nama obat kedalam *list* dapat dilihat pada gambar 5.6 berikut.

```
tambah.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        HashMap<String, String> item = new HashMap<>();
        item.put(TAG_NAMA_OBAT, nama_obat.getText().toString());
        item.put(TAG_JUMLAH, jumlah_obat.getText().toString());
        itemList.add(item);
        adapter.notifyDataSetChanged();
        nama_obat.setText("");
        jumlah_obat.setText("");
    }
});
```

Gambar 5.6 Perintah yang dijalankan pada tombol tambah

Setelah semua obat yang dibutuhkan masuk kedalam daftar pesanan, sales dapat menekan tombol “centang” yang ada pada *menu bar* aplikasi. Tombol tersebut akan menjalankan fungsi untuk membuka *class* yang menampilkan pengisian informasi pemesan. Daftar obat pesanan yang disimpan pada *arraylist* akan disalurkan melalui proses *intent*. Gambar 5.7 adalah kode program yang ada pada tombol “Done”.

```
Intent i = new Intent(tambahPesanan.this, tambahPesananCont.class);
i.putExtra("daftar_obat", (ArrayList<HashMap<String,String>>)itemList);
startActivity(i);
```

Gambar 5.7 Kode program pada tombol “Done”

Pada halaman informasi pemesan, sales akan memasukkan nama dan alamat apotek/rumah sakit pemesan. Pada *class* ini dimasukkan google maps untuk melakukan pengisian alamat secara otomatis, serta mengambil titik koordinat lokasi agar dapat digunakan untuk menampilkan alamat pemesan pada aplikasi kurir. Gambar 5.8 merupakan tampilan dari halaman informasi pemesan, sebuah *marker* akan muncul ketika sales memilih suatu lokasi.



Gambar 5.8 Halaman informasi pemesan

Pengambilan alamat secara otomatis dilakukan dengan memanfaatkan *geocoder* yang merupakan fitur yang disediakan oleh google maps. Kode program yang digunakan untuk mengubah titik koordinat menjadi teks alamat dapat dilihat pada gambar 5.9 berikut.

```
private String getAddressFromLatLng(LatLng latLng) {
    Geocoder geocoder = new Geocoder(getApplicationContext());

    String address = "";
    try {
        address = geocoder
            .getFromLocation( latLng.latitude, latLng.longitude, 1 )
            .get( 0 ).getAddressLine( 0 );
    } catch (IOException e ) {
    }

    return address;
}
```

Gambar 5.9 Mengubah titik koordinat menjadi teks alamat

Setelah semua informasi pemesan dimasukkan, sales cukup menekan tombol “centang” untuk menyimpan pesanan tersebut. Aplikasi web yang ada pada server akan menjalankan fungsi untuk menyimpan data pesanan ke dalam *database*.

## 5.4 Pembuatan Aplikasi Kurir Obat

Pada bagian ini akan dijelaskan pembuatan aplikasi android kurir obat dalam bentuk kode program yang digunakan sesuai dengan desain aplikasi.

### 5.4.1 Fungsi *Login*

*Login* kedalam aplikasi ini menggunakan proses yang sama dengan fungsi *login* pada aplikasi sales obat. Aplikasi membutuhkan kombinasi *username* dan *password* untuk melakukan proses *login*. Jika kombinasi *username* dan *password* sesuai dengan data yang ada pada *database*, maka

aplikasi akan menjalankan *class* yang menampilkan halaman *home* aplikasi.

## 5.4.2 Fungsi Menampilkan Lokasi Pemesan

Aplikasi kurir obat ini bertujuan untuk memudahkan kurir dalam menemukan lokasi dari pemesan obat. Oleh karena itu, aplikasi ini dilengkapi dengan google maps untuk menampilkan lokasi pemesan obat. Google maps akan diinisialisasi pada suatu posisi koordinat dengan memanfaatkan *LocationService* dan mengambil data dari posisi terakhir ketika GPS diaktifkan. Jika *LocationService* tidak memiliki data posisi terakhir, maka untuk menghindari *error* pada aplikasi google maps akan diinisialisasi pada koordinat kota Surabaya. Kode program untuk menginisialisasi google map dapat dilihat pada gambar 5.10 berikut.

```
mCurrentLocation = LocationServices
    .FusedLocationApi
    .getLastLocation(mGoogleApiClient);
if(mCurrentLocation != null) {
    initCamera(mCurrentLocation);
} else {
    mCurrentLocation = new Location("");
    mCurrentLocation.setLatitude(-7.26422404);
    mCurrentLocation.setLongitude(112.75165558);
    initCamera(mCurrentLocation);
}
```

Gambar 5.10 Kode program inisialisasi lokasi google maps

Rute yang dapat ditempuh oleh kurir ke lokasi pengirim dapat ditampilkan dalam google maps. Fitur ini dibuat dengan menggunakan *Library Google Direction Android* yang dikembangkan dengan lisensi *open source*. *Library* ini dapat diakses pada website <https://github.com/jd-alexander/Google-Directions-Android>. Rute secara otomatis dibuat dari posisi kurir ke lokasi pemesan. Kode program ini dimasukkan kedalam tombol “*MyLocation*” yang ada pada google maps.

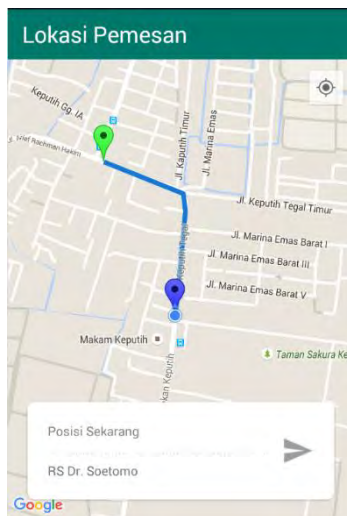


Gambar 5.11 merupakan kode program yang digunakan untuk menggunakan *Library Google Directions* Android.

```
public void route() {
    progressDialog = ProgressDialog.show(this, "Please wait.",
        "Fetching route information.", true);
    Routing routing = new Routing.Builder()
        .travelMode(AbstractRouting.TravelMode.DRIVING)
        .withListener(this)
        .alternativeRoutes(true)
        .waypoints(titik_awal, titik_akhir)
        .build();
    routing.execute();
}
```

Gambar 5.11 Kode program untuk membuat rute perjalanan

Setelah tombol “*MyLocation*” ditekan, aplikasi akan menampilkan dua buah *marker* berwarna biru dan hijau. *Marker* biru merupakan lokasi kurir dan hijau merupakan lokasi pemesan. Gambar 5.12 menunjukkan contoh tampilan rute yang dihasilkan aplikasi.



Gambar 5.12 Contoh rute dari posisi kurir ke lokasi pemesan

Rute yang ditampilkan dengan menggunakan *library* ini bisa sampai lima jalur alternatif, dengan demikian kurir dapat diberikan kemudahan dalam memilih rute yang ingin dilalui. Google maps membutuhkan GPS untuk menentukan posisi kurir, oleh karena itu ketika akan menampilkan halaman google maps ini aplikasi akan melakukan pengecekan terlebih dahulu apakah GPS dari *mobile phone* sudah dalam keadaan aktif atau tidak. Kode program yang digunakan untuk melakukan pengecekan tersebut dapat dilihat pada gambar 5.13.

```

if (statusNetwork) {
    mGoogleApiClient.connect();
} else if (statusGPS) {
    mGoogleApiClient.connect();
} else {
    AlertDialog.Builder promptgps = new AlertDialog.Builder(this);
    promptgps.setMessage("Activate GPS !")
        .setCancelable(false)
        .setTitle("Need GPS Permission")
        .setPositiveButton("Ok",
            (OnClickListener) (dialog, id) → {
                Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                startActivityForResult(intent, 0);
            })
        .setNegativeButton("No",
            (OnClickListener) (dialog, id) → {
                Lokasi.this.finish();
            });
    //Showing the alert dialog
    AlertDialog showprompt = promptgps.create();
    promptgps.show();
}

```

Gambar 5.13 Pengecekan status GPS

GPS yang tersedia pada *mobile phone* memiliki tiga metode penentuan lokasi, yaitu melalui koneksi ke jaringan *provider*, *gps device*, dan penggabungan keduanya. Oleh karena itu pengecekan dilakukan pada status jaringan *provider* dan status *device* GPS.

Jika status jaringan dan *gps device* tidak dalam keadaan aktif, maka aplikasi akan menampilkan pesan agar kurir mengaktifkan status *gps* pada *mobile phone*.

### 5.4.3 Fungsi Foto Dokumentasi

Fungsi foto dokumentasi digunakan untuk menunjukkan bahwa pesanan sudah sampai ke pemesan. Aplikasi akan dilengkapi dengan kode program untuk memanggil kamera dari *mobile phone* untuk mengambil foto. Gambar 5.14 merupakan kode program untuk memanggil kamera melalui *intent*.

```
Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
// Ensure that there's a camera activity to handle the intent
if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
    // Create the File where the photo should go
    File photoFile = null;
    try {
        photoFile = createImageFile();
    } catch (IOException ex) {
        // Error occurred while creating the File
    }
    // Continue only if the File was successfully created
    if (photoFile != null) {
        takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT,
            Uri.fromFile(photoFile));
        startActivityForResult(takePictureIntent, PICK_IMAGE_REQUEST);
    }
}
```

Gambar 5.14 *Intent* memanggil kamera

*Intent* untuk memanggil kamera tersebut dijalankan sebagai *activityforresult*, dengan cara tersebut maka foto yang diambil melalui kamera dapat digunakan pada aplikasi. Foto dokumentasi akan disimpan terlebih dahulu menjadi file *temporary*. Aplikasi akan mengambil *path* dari foto tersebut kemudian menampilkannya kedalam *imageview*. Kode program untuk menampilkan foto dokumentasi kedalam aplikasi dapat dilihat pada gambar 5.15.

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK) {
        try {
            bitmap = MediaStore.Images.Media.getBitmap(this.getContentResolver(),
                Uri.parse(mCurrentPhotoPath));
            imageView.setImageBitmap(bitmap);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Gambar 5.15 *ActivityResult* untuk menampilkan foto

Foto dokumentasi dikirim ke server dengan mengubahnya terlebih dahulu kedalam *bytearray* menggunakan *bytearrayoutputstream*. Kode program untuk mengubah file foto kedalam *bytearray* dapat dilihat pada gambar 5.16.

```

public String getStringImage(Bitmap bmp){
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    bmp.compress(Bitmap.CompressFormat.JPEG, 50, baos);
    byte[] imageBytes = baos.toByteArray();
    String encodedImage = Base64.encodeToString(imageBytes, Base64.DEFAULT);
    return encodedImage;
}

```

Gambar 5.16 Mengubah file foto kedalam *bytearray*

Hasil pengubahan file foto kedalam *bytearray* akan dienkripsi menjadi tipe data *String*. Data tersebut kemudian dikirim ke server dengan menggunakan *library volley*. *Stringrequest* akan digunakan sebagai metode pengiriman data, karena data enkripsi foto sudah diubah kedalam tipe data *String*. Kode program yang digunakan untuk mengirim data ke server dapat dilihat pada gambar 5.17.

```
StringRequest stringRequest = new StringRequest(Request.Method.POST, UPLOAD_URL,
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            uploading.dismiss();
            Toast.makeText(getApplicationContext(), response, Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            uploading.dismiss();
            Toast.makeText(getApplicationContext(), error.getMessage().toString(),
                Toast.LENGTH_SHORT).show();
        }
    })
```

Gambar 5.17 *Stringrequest* dengan metode *POST*

Data foto yang sudah dienkrpsi dimasukkan kedalam parameter *request* dan dikirim dengan metode *POST*. Gambar 5.18 merupakan parameter yang digunakan.

```
@Override
protected Map<String, String> getParams() throws AuthFailureError {
    //Converting Bitmap to String
    String image = getStringImage(bitmap);

    String id;
    Intent i = getIntent();
    id = i.getStringExtra("id");

    //Creating parameters
    Map<String,String> params = new Hashtable<>();

    //Adding parameters
    params.put(KEY_IMAGE, image);
    params.put(KEY_ID, id);

    //returning parameters
    return params;
}
```

Gambar 5.18 Parameter *Stringrequest*

File php yang dipanggil melalui *request* ini akan memproses data menjadi file foto. Setelah proses upload foto selesai, maka kode program dalam file php yang dijalankan juga akan mengubah status pesanan menjadi “terkirim”. Gambar 5.19 merupakan kode program yang digunakan untuk memproses data menjadi file gambar.

```

if($_SERVER['REQUEST_METHOD']=='POST'){
    $id= $_POST['id'];
    $image = $_POST['image'];
    $status="done";
    require_once('conection.php');
    $path = "bukti/$id.png";
    $actualpath      =      "http://192.168.1.22/cour-
monitor/android/$path";
    $query = $con->prepare("UPDATE orderan SET
bukti=? , status=? where id_pesanan=?");
    $query->bind_param('sss',
    $actualpath,$status,$id);
    if($query->execute()){
        file_put_contents($path,base64_decode($image));
        echo "Successfully Uploaded";
    }
}

```

Gambar 5.19 Kode program untuk menyimpan foto

#### 5.4.4 Fungsi *Push Notification*

Fitur *push notification* digunakan ketika kurir mendapat tugas untuk mengirim suatu pesanan. Fitur ini akan dibuat dengan menggunakan *Google Cloud Messaging* (GCM). GCM merupakan layanan dari google untuk mengirim pesan antara *mobile phone* dengan server. Layanan ini dapat digunakan dengan mendaftarkan *project* aplikasi di halaman website <https://developers.google.com/cloud-messaging/>. Untuk aplikasi android, dibutuhkan penggunaan *library* google *playservice*. Pengembangan aplikasi dengan menggunakan Android SDK dapat mengimplementasikannya dengan menambahkan *library* tersebut pada file *gradle*.

GCM membutuhkan informasi *device mobile phone* berupa *token* agar dapat mengirim pesan sesuai dengan tujuan. Oleh karena itu dibutuhkan *class* yang berfungsi untuk melakukan

registrasi *device* untuk mendapatkan *token* tersebut. Kode program yang digunakan untuk mendapatkan *token* dapat dilihat pada gambar 5.20.

```
InstanceID instanceID = InstanceID.getInstance(getApplicationContext());

//Getting the token from the instance id
token = instanceID.getToken(getString(R.string.gcm_defaultSenderId),
    GoogleCloudMessaging.INSTANCE_ID_SCOPE, null);
```

Gambar 5.20 Registrasi kode *token*

Notifikasi pesan baru harus dapat ditampilkan walaupun aplikasi tidak sedang dijalankan. Untuk itu diperlukan suatu *class* yang berfungsi sebagai *service*, *service* inilah yang akan berjalan dibelakang layar untuk menerima pesan yang dikirim dari server. Kode program yang digunakan untuk membuat *receiver service* GCM dapat dilihat pada gambar 5.21 berikut.

```
Intent intent = new Intent(this, MainActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
int requestCode = 0;
PendingIntent pendingIntent = PendingIntent.getActivity(this, requestCode,
    intent, PendingIntent.FLAG_ONE_SHOT);
Uri sound = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
NotificationCompat.Builder noBuilder = new NotificationCompat.Builder(this)
    .setSmallIcon(R.mipmap.ic_launcher)
    .setContentText(message)
    .setTicker(message)
    .setContentTitle("App Kurir")
    .setAutoCancel(true)
    .setSound(sound)
    .setContentIntent(pendingIntent);
```

Gambar 5.21 *Receiver service* aplikasi

Pesan yang diterima dari server akan ditampilkan melalui notifikasi pada *mobile phone*. Konfigurasi notifikasi dilakukan pada *class receiver service* ini, seperti *icon*, *title*, *sound*, serta *message* dari notifikasi.

Pada aplikasi distributor obat, pesan notifikasi dikirimkan setelah distributor menentukan kurir yang bertugas untuk mengirim pesan tersebut. Aplikasi distributor obat juga

membutuhkan kode registrasi yang digunakan untuk menggunakan layanan GCM. Kode registrasi tersebut didapatkan pada saat mendaftar di layanan GCM.

Kode program yang digunakan untuk mengirim pesan notifikasi dapat dilihat pada gambar 5.22 berikut.

```
$headers = array
(
    'Authorization: key=' . $api_key,
    'Content-Type: application/json'
);
$ch = curl_init();
curl_setopt( $ch,CURLOPT_URL,
'https://android.googleapis.com/gcm/send' );
curl_setopt( $ch,CURLOPT_POST, true );
curl_setopt( $ch,CURLOPT_HTTPHEADER, $headers );
curl_setopt( $ch,CURLOPT_RETURNTRANSFER, true );
curl_setopt( $ch,CURLOPT_SSL_VERIFYPEER, false );
curl_setopt( $ch,CURLOPT_POSTFIELDS, json_encode(
$fields ) );
$result = curl_exec($ch );
curl_close( $ch );
```

Gambar 5.22 Pesan notifikasi dari server

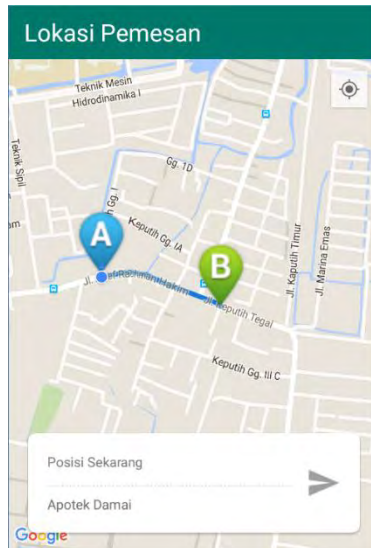
Pesan terlebih dahulu dikirim ke server layanan GCM dengan menggunakan *client url library (curl)*. GCM kemudian mengirimkan pesan tersebut ke aplikasi kurir obat sesuai dengan kode *token*.

### 5.4.5 Fungsi *Track* Lokasi Kurir

Lokasi kurir yang sedang mengantarkan pesanan dapat diketahui dengan memanfaatkan fitur GPS yang ada pada *mobile phone*. Lokasi kurir akan disimpan secara berkala ketika kurir mengakses lokasi pemesan. *Track* lokasi dimulai setelah rute



jalan dipilih. Gambar 5.23 merupakan contoh rute yang ditampilkan oleh google maps pada aplikasi.



Gambar 5.23 Tampilan rute pada google maps

Perubahan lokasi kurir dapat diketahui dengan menggunakan *location listener*. Pada *Method onlocationchanged* dimasukkan kode program yang berfungsi untuk mengubah posisi kamera map agar mengikuti titik koordinat kurir. Setiap perubahan koordinat akan disimpan kedalam database SQLite. Kode program yang digunakan pada *method onlocationchanged* dapat dilihat pada gambar 5.24.

```
public void onLocationChanged(Location location) {
    double lat = location.getLatitude();
    double lng = location.getLongitude();
    CameraUpdate center = CameraUpdateFactory.newLatLng(new LatLng(lat,lng));
    CameraUpdate zoom = CameraUpdateFactory.zoomTo(16);
    mMap.moveCamera(center);
    mMap.animateCamera(zoom);
    gpstrack.insertlocation(id,String.valueOf(lat),String.valueOf(lng),user, date,"no");
}
```

Gambar 5.24 *Method onlocationchanged*

Data yang disimpan kedalam database SQLite yaitu titik *latitude*, *longitude*, *user* kurir, tanggal *tracking* lokasi, dan status sinkronisasi data. Data tersebut yang akan ditampilkan pada aplikasi web. Status sinkronisasi yang dimasukkan kedalam database SQLite berguna untuk menandakan apakah setiap rekaman titik koordinat sudah disimpan ke server atau belum.

*Method insertlocation* yang digunakan untuk menyimpan data dipanggil dari dalam *class DBHelper* yang berfungsi untuk mengelola database SQLite. Kode program yang mengelola database untuk menyimpan hasil *tracking* lokasi kurir dapat dilihat pada gambar 5.25 berikut.

```
public GPSDatabase(Context context, String name, SQLiteDatabase.CursorFactory factory, int
    version) {
    super(context, name, factory, version);
}

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table location(id integer primary key autoincrement, " +
        "id_pesanan integer(5), lat float(10,6)," +
        "lng float(10,6), kurir varchar(20), tgl date, synced varchar(5));");
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS location;");
    onCreate(db);
}

public void insertlocation(String id_pesanan,String lat, String lng, String kurir, String tgl,
    String synced){
    ContentValues contentValues = new ContentValues();
    contentValues.put("id_pesanan", id_pesanan);
    contentValues.put("lat", lat);
    contentValues.put("lng", lng);
    contentValues.put("kurir", kurir);
    contentValues.put("tgl", tgl);
    contentValues.put("synced", synced);
    this.getWritableDatabase().insert("location", null, contentValues);
}
```

Gambar 5.25 *Class DBHelper* untuk mengelola db SQLite

Setelah data lokasi berhasil disinkronisasi dengan server, maka status setiap *record* akan diubah menjadi “yes”. Kode program

yang dijalankan untuk mengubah status tersebut dapat dilihat pada gambar 5.26 berikut.

```
public void updateSyncStatus() {
    this.getWritableDatabase().execSQL("update location set synced='yes' where synced='no'");
    this.getWritableDatabase().close();
}
```

Gambar 5.26 Pengubahan status sinkronisasi data

## 5.5 Pembuatan Aplikasi Distributor Obat

Pada bagian ini akan dijelaskan pembuatan aplikasi distributor obat dalam bentuk kode program yang digunakan sesuai dengan desain aplikasi.

### 5.5.1 Fungsi *Login*

*Login* kedalam aplikasi distributor obat dibuat dengan menggunakan kombinasi *username* dan *password*. Kombinasi tersebut akan dicek kesesuaiannya dengan data yang ada pada database. Jika kombinasi *username* dan *password* tersebut ada pada database, maka aplikasi akan menampilkan halaman *home* aplikasi. Kode program yang digunakan untuk validasi kombinasi *username* dan *password* dapat dilihat pada gambar 5.27 berikut.

```
$query = $con->prepare("SELECT * FROM user WHERE
    BINARY user=? AND BINARY password=? AND level=?");
$query->bind_param('ssi',$user,$password,$lev);
$query->execute();
$query->bind_result($id,$use,$passe,$level);
$check = $query->fetch();
if(isset($check)){
    $_SESSION['user']=$use;
    header("location:../");
}
```

Gambar 5.27 Validasi *username* dan *password*

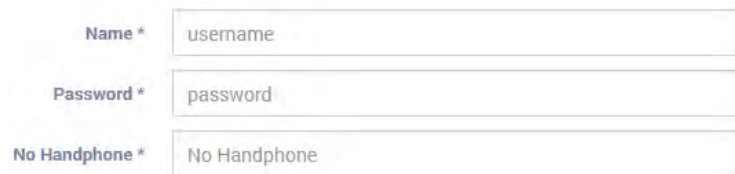
*SQLpreparedstatement* digunakan dalam menjalankan *query* untuk meningkatkan keamanan aplikasi. *Query* tidak akan

terpengaruh dengan data yang dimasukkan, sehingga penggunaan teknik *sqlinjection* sebagai salah satu cara untuk melewati validasi data dapat dicegah.

Parameter *binary* digunakan untuk meningkatkan sensitifitas terhadap karakter yang dimasukkan, sehingga penggunaan huruf kapital dan kecil dapat dibedakan.

### 5.5.2 Fungsi Pengelola *User*

*User* dari setiap aplikasi android baik itu kurir obat maupun sales obat, dikelola melalui aplikasi distributor obat ini. Pengelolaan *user* untuk kurir dan sales dibedakan dengan tujuan agar aplikasi distributor obat lebih *user friendly*. Penambahan *user* untuk kurir dan sales hanya membutuhkan data dasar seperti *username*, *password*, dan *no handphone*. Tampilan *form* penambahan *user* dapat dilihat pada gambar 5.28 berikut.



The image shows a web form for adding a user. It consists of three input fields stacked vertically. The first field is labeled 'Name \*' and contains the text 'username'. The second field is labeled 'Password \*' and contains the text 'password'. The third field is labeled 'No Handphone \*' and contains the text 'No Handphone'.

Gambar 5.28 *Form* penambahan *user*

Data akan disimpan kedalam database dengan menggunakan kode program yang ditampilkan pada gambar 5.29 berikut.

```
$query = $con->prepare("INSERT INTO user
(user,password,level) VALUES (?,?,?)");
$query->bind_param('sss',$user,$password,$lvl);
$user = $_POST['username'];
$password = md5($_POST['password']);
$lvl="2";
$query->execute();
```

Gambar 5.29 Penyimpanan *user* baru kedalam database


Penambahan *level* pada data *user* bertujuan untuk membedakan status dari *user* tersebut, apakah sebagai kurir atau sales.

### 5.5.3 Fungsi Pengelola Pesanan

Aplikasi distributor obat memiliki fungsi untuk pengelolaan pesanan. Setiap pesanan baik yang baru dibuat, dalam status pengiriman, maupun yang sudah dikirim dapat ditampilkan oleh aplikasi ini. Setiap data pesanan akan ditampilkan dalam halaman yang berbeda sesuai dengan status pesanan tersebut. Berikut adalah penjelasan setiap halaman pengelolaan pesanan.

#### 1. Halaman Pesanan Baru

Pesanan yang baru dibuat oleh sales akan ditampilkan pada halaman ini. Kurir yang bertugas untuk mengirim pesanan akan dikelola melalui halaman ini. Tampilan penentuan kurir yang bertugas dapat dilihat pada gambar 5.30 berikut.



The image shows a web form titled "Penerima". It contains a text input field with the value "RS Priangan". Below this is a section titled "Pilih Kurir" which contains a dropdown menu with the text "Pilih" and a small downward arrow. At the bottom of the form is a green button with the text "Simpan".

Gambar 5.30 Tampilan penentuan kurir

Setelah kurir ditentukan kemudian tombol “simpan” ditekan, maka aplikasi akan menjalankan kode program yang berfungsi untuk menyimpan status tersebut. Pada proses ini juga pengiriman notifikasi pesanan ke aplikasi kurir dilakukan. Kode program yang berfungsi untuk menyimpan informasi penentuan kurir dapat dilihat pada gambar 5.31.

```

$query = $con->prepare("UPDATE orderan SET kurir=?,
status=? WHERE id_pesanan=?");
$query->
bind_param('sss',$_POST['user_kurir'],$status,$_POST['id_orde
r']);
$status="proses";
$query->execute();

```

Gambar 5.31 Simpan informasi penentuan kurir

## 2. Halaman Pesanan dalam Proses Pengiriman

Pesanan baru yang sudah ditentukan kurir untuk mengirim pesanan tersebut akan ditampilkan pada halaman ini. Daftar pesanan yang dalam status proses pengiriman ditampilkan dalam tabel dengan informasi nama pemesan, alamat, serta kurir yang bertugas untuk mengantarkan pesanan tersebut. Gambar 5.35 merupakan tampilan tabel berisi daftar pesanan dalam proses pengiriman.

Pemesan	Alamat	Kurir
Apotek Bahagia	Jalan Keputih Tegal No.3 A	kurir1
Apotek Capjay	keputih	kurir1
Apotek Damai	Jalan Keputih Tegal No.9	kurir1
Apotek Gebang	keputih	kurir1

Gambar 5.32 Daftar pesanan dalam proses pengiriman

Kode program yang digunakan untuk menampilkan daftar pesanan tersebut dapat dilihat pada gambar 5.36 berikut.

```

$query = $con->prepare("SELECT * FROM orderan where
status='proses'");
$query->execute();
$query->bind_result($id,$penerima,$kurir,$status,$alamat,$bukti);

```

Gambar 5.32 Mengambil data pesanan dalam proses pengiriman

### 3. Halaman Pesanan Selesai

Pesanan yang sudah dikirimkan oleh kurir akan ditampilkan pada halaman ini. Daftar pesanan yang sudah selesai dikirim ditampilkan dalam tabel dengan informasi nama pemesan, alamat, kurir yang bertugas, serta foto dokumentasi pengiriman yang di-*upload* oleh kurir. Tabel berisi daftar pesanan selesai dapat dilihat pada gambar 5.33.

Pemesan	Alamat	Kurir	Foto
Apotek Bahagia	Keputih Perintis 2	kurir1	
apotek budi	keputih	kurir1	
Apotek Dajen	keputih	user1	
Apotek Gebang	Gebang Lor no.25	user1	
Apotek Jenfa	keputih	kurir1	

Gambar 5.34 Daftar pesanan selesai

Kode program yang digunakan untuk menampilkan daftar pesanan selesai dapat dilihat pada gambar 5.38 berikut.

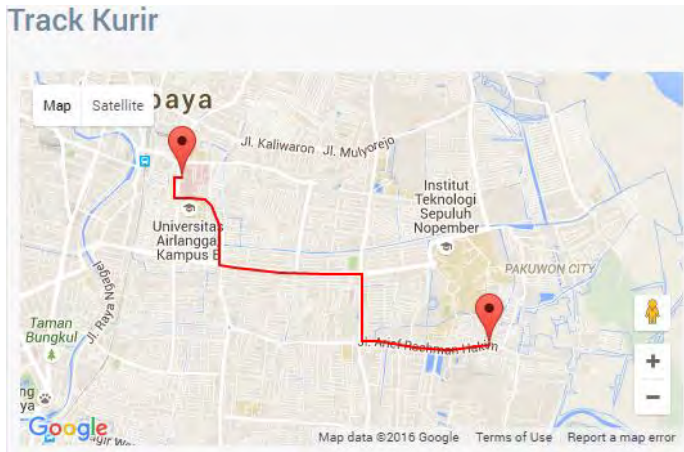
```
$query = $con->prepare("SELECT * FROM orderan where
status=done");
$query->execute();
$query->bind_result($id,$penerima,$skurir,$status,$alamat,$bukti);
```

Gambar 5.34 Mengambil data pesanan selesai

#### 5.5.4 Fungsi *Track Kurir*

Lokasi kurir yang di-*track* melalui aplikasi kurir obat kemudian disimpan ke database server akan disajikan menggunakan google maps. Setiap titik koordinat yang disimpan akan dihubungkan menjadi suatu garis. Kurir yang akan dilihat hasil *track* lokasinya dipilih terlebih dahulu, kemudian diikuti

dengan pemilihan tanggal *tracking* lokasi. Jika aplikasi menemukan data sesuai dengan parameter yang ditentukan, maka google maps akan ditampilkan beserta garis yang dibentuk menggunakan fungsi *polyline* berdasarkan data koordinat yang dimiliki. Tampilan halaman *track* kurir dapat dilihat pada gambar 5.35.



Gambar 5.36 Tampilan halaman *track* kurir

Tahapan yang dilalui untuk menampilkan hasil *track* lokasi kurir tersebut dijelaskan sebagai berikut.

1. Mengambil data dari database

Aplikasi terlebih dahulu mengambil semua data sesuai dengan parameter yang digunakan, yakni berdasarkan *user* kurir dan tanggal *tracking* lokasi. Kode program yang digunakan untuk mengambil data tersebut dapat dilihat pada gambar 5.37.



```

$query = $con->prepare("SELECT * FROM track where kurir=? and
tanggal=?");
$query -> bind_param('ss',$ _GET['kurir'],$ _GET['tgl']);
$query->execute();
$query->store_result();
$query->bind_result($id,$id_pesanan,$lat,$lng,$skurir,$tanggal);

```

Gambar 5.37 Mengambil data *track* lokasi

## 2. Inisialisasi google maps

Setelah data hasil *track* berhasil diambil dari database, aplikasi kemudian menginisiasi google maps dengan beberapa parameter yang dibutuhkan. Kode program yang digunakan dapat dilihat pada gambar 5.38 berikut.

```

var map = new
google.maps.Map(document.getElementById('map_canvas'), {
    zoom: 13,
    center: {lat: -7.257472, lng: 112.752088},
    mapTypeId: google.maps.MapTypeId.TERRAIN
});

```

Gambar 5.38 Inisialisasi google map

## 3. Menggambar garis berdasarkan data lokasi

Data lokasi kemudian digambar menggunakan fungsi *polyline* dari google map. Setiap titik koordinat dihubungkan sehingga membentuk garis yang menunjukkan pergerakan kurir. Kode program yang digunakan untuk membentuk garis tersebut dapat dilihat pada gambar 5.39 berikut.

```

var trackpath = new google.maps.Polyline({
    path: track,
    geodesic: true,
    strokeColor: '#FF0000',
    strokeOpacity: 1.0,
    strokeWeight: 2,
    map: map
});

```

Gambar 5.39 Membuat garis dengan *polylines*

Garis yang digambar pada peta menunjukkan perubahan posisi kurir ketika mengirim pesanan. Semua hasil *track* lokasi akan digabung menjadi satu garis yang memiliki titik awal dan titik akhir yang ditandai dengan *marker* pada google maps.

## 5.6 Pengujian Aplikasi

Pada bagian ini akan dibahas pengujian yang dilakukan pada aplikasi, termasuk didalamnya pengujian sistem dan pengujian kompatibilitas aplikasi sesuai dengan versi android API yang ditargetkan.

### 5.6.1 Pengujian Fungsional

Pengujian fungsional yaitu melakukan uji coba terhadap aplikasi secara keseluruhan terkait spesifikasi aplikasi yang telah ditentukan diawal. Setiap fungsi dari aplikasi akan dicoba, termasuk fitur yang menggunakan layanan dari google seperti GCM dan Maps. Pengujian dilakukan dengan menggunakan format test case seperti pada tabel 5.4.

Tabel 5.4 Format Test Case

No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Aksi yang dilaku	Data yang dimasuk	Hasil yang terjadi pada aplikasi	Hasil yang dilihat oleh pengguna	Sukses

	kan aktor	kan aktor			
--	--------------	--------------	--	--	--

### 5.6.2 Pengujian Kompatibilitas Aplikasi Android

Aplikasi android yang dibuat untuk kurir dan sales akan diuji kompatibilitasnya sesuai dengan target minimum dan maximum versi android API yang ditentukan diawal. Versi API minimum yang ditargetkan adalah 19 dan versi terbaru yang ditargetkan adalah 23. Setiap aplikasi akan dicoba pada kedua versi android API tersebut untuk mengetahui apakah aplikasi dapat berjalan dengan baik pada *range* target API tersebut.

## **BAB VI**

### **HASIL DAN PEMBAHASAN**

Pada bab ini akan dijelaskan hasil dari pengujian aplikasi serta pembahasan terhadap hasil pengujian yang dilakukan. Berikut adalah penjabaran untuk hasil dan pembahasan pengujian yang dilakukan.

#### **6.1 Hasil**

Pengujian yang dilakukan pada aplikasi dijabarkan pada sub bab ini. Detail pengujian fungsional dapat dilihat pada lampiran A.

##### **6.1.1 Pengujian Fungsional Aplikasi Sales Obat**

Hasil dari pengujian fungsional yang dilakukan terhadap aplikasi sales obat dapat dilihat pada tabel 6.1.

Tabel 6.1 Hasil uji coba fungsional aplikasi sales obat

<b>Kode Test Case</b>	<b>Nama Test Case</b>	<b>Status</b>
TC-01	<i>Login</i>	Terpenuhi
TC-02	Tambah pesanan obat baru	Terpenuhi
TC-03	Simpan lokasi pemesan	Terpenuhi
TC-04	Melihat detail pesanan	Terpenuhi
TC-05	<i>Logout</i>	Terpenuhi

##### **6.1.2 Pengujian Fungsional Aplikasi Kurir Obat**

Hasil dari pengujian fungsional yang dilakukan terhadap aplikasi kurir obat dapat dilihat pada tabel 6.2.

Tabel 6.2 Hasil uji coba fungsional aplikasi kurir obat

<b>Kode Test Case</b>	<b>Nama Test Case</b>	<b>Status</b>
TC-06	<i>Login</i>	Terpenuhi
TC-07	Melihat detail pesanan	Terpenuhi
TC-08	Melihat lokasi pemesan	Terpenuhi
TC-09	Mengupload foto dokumentasi	Terpenuhi
TC-10	<i>Push notification</i>	Terpenuhi
TC-11	<i>Track</i> lokasi	Terpenuhi
TC-12	<i>Logout</i>	Terpenuhi

### 6.1.3 Pengujian Fungsional Aplikasi Distributor Obat

Hasil dari pengujian fungsional yang dilakukan terhadap aplikasi kurir obat dapat dilihat pada tabel 6.3.

Tabel 6.3 Hasil uji coba fungsional aplikasi distributor obat

<b>Kode Test Case</b>	<b>Nama Test Case</b>	<b>Status</b>
TC-13	<i>Login</i>	Terpenuhi
TC-14	Mengelola pesanan	Terpenuhi
TC-15	Mengelola user	Terpenuhi
TC-16	Melihat foto dokumentasi	Terpenuhi
TC-17	Melihat hasil <i>track</i> lokasi kurir	Terpenuhi
TC-18	<i>Logout</i>	Terpenuhi

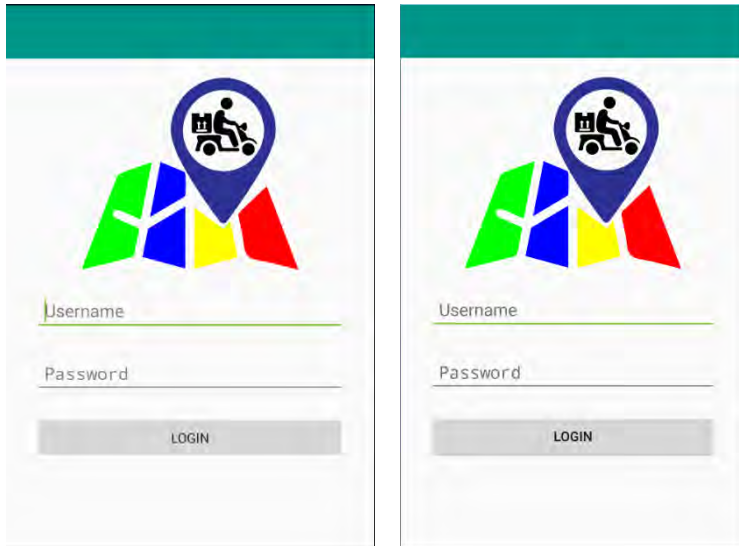
### 6.1.4 Pengujian Kompatibilitas Aplikasi Android

Aplikasi android untuk kurir dan sales dibangun dengan target android API minimal versi 19 sampai API versi 23. Setiap versi android API tentu memiliki perbedaan, oleh karena itu diperlukan pengujian terkait kompatibilitas aplikasi pada range API tersebut. Pengujian akan dilakukan dengan bantuan *software android emulator Genymotion* versi 2.6.0. *Software* ini mampu meng-*emulate* sistem operasi android dengan berbagai versi API, dengan demikian dapat dibuat dua buah *emulator* dengan versi API 19 dan 23.

Pengujian aplikasi android sales obat dan kurir obat dilakukan dengan menjalankannya pada kedua versi API. Dokumentasi pengujian dilakukan dengan menunjukkan tampilan aplikasi pada kedua versi API. Tampilan aplikasi pada API versi 19 diletakkan disebelah kiri gambar, dan API versi 23 pada sebelah kanan. Penjabaran pengujian dijelaskan sebagai berikut.

#### 1. *Login* aplikasi

Aplikasi diharapkan dapat menampilkan *form username, password*, serta tombol dengan label “*Login*”. Tampilan halaman *login* pada kedua versi API dapat dilihat pada gambar 6.1 berikut.



Gambar 6.1 Tampilan *login* aplikasi

## 2. Daftar pesanan

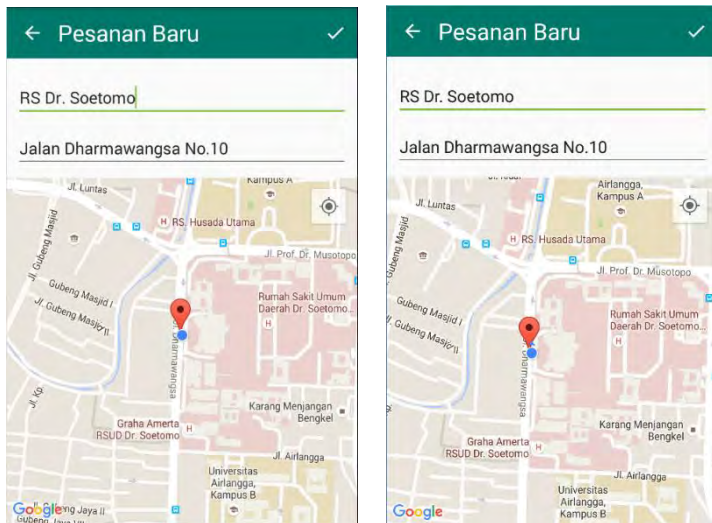
Aplikasi diharapkan dapat menampilkan daftar pesanan yang sudah disimpan didatabase server, kedalam *tab* sesuai dengan status dari pesanan tersebut. Tampilan daftar pesanan pada *home* aplikasi dapat dilihat pada gambar 6.2.



Gambar 6.2 Tampilan daftar pesanan

## 3. Buat pesanan baru

Aplikasi diharapkan dapat menampilkan *form* obat pesanan pada halaman penambahan obat pesanan dan *form* informasi pemesan serta google maps untuk mengambil titik koordinat lokasi pada halaman informasi pemesan. Tampilan halaman informasi pemesan pada kedua versi API dapat dilihat pada gambar 6.3.

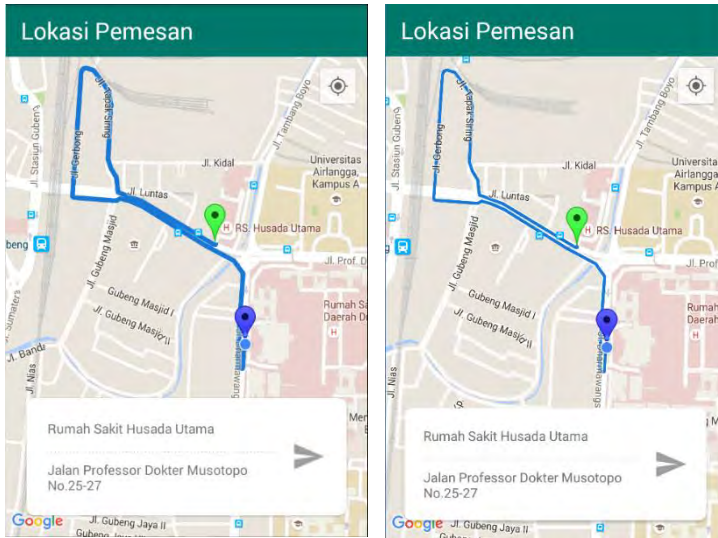


Gambar 6.3 Pengisian informasi pemesan

#### 4. Lokasi pemesan

Aplikasi kurir obat diharapkan dapat menampilkan lokasi pemesan pada google maps, dapat menampilkan nama pemesan serta alamat pemesan, serta dapat melakukan fungsi untuk menampilkan rute perjalanan dari posisi kurir ke tempat pemesan. Tampilan lokasi pemesan serta contoh rute perjalanan menuju lokasi pemesan dapat dilihat pada gambar 6.4.

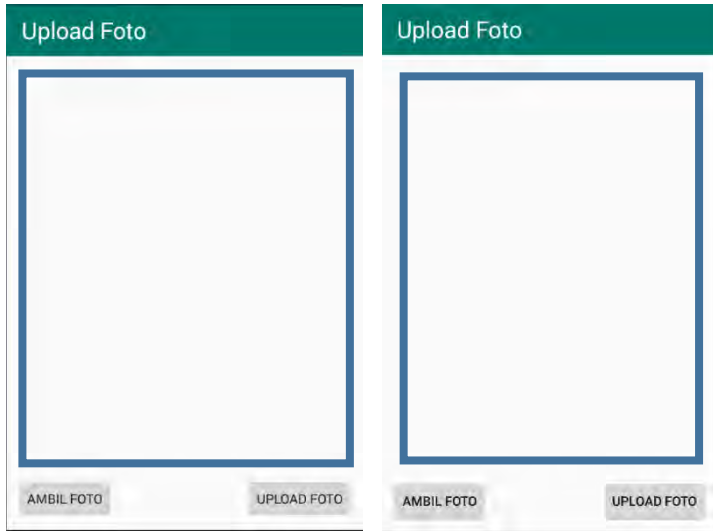




Gambar 6.4 Rute perjalanan menuju lokasi pemesan

## 5. *Upload* foto

Setelah kurir sampai ke lokasi pemesan dan akan melakukan dokumentasi foto penyerahan pesanan, aplikasi diharapkan dapat menampilkan tombol “ambil foto” dan “*upload* foto”, serta dapat melakukan fungsinya masing-masing dengan baik. Tampilan halaman *upload* foto dapat dilihat pada gambar 6.5. Foto yang diambil dari kamera akan ditampilkan dibagian tengah aplikasi, area tempat foto ditandai dengan kotak biru pada gambar.



Gambar 6.5 *Upload* foto dokumentasi

## 6.2 Pembahasan

Pada sub bab akan dijelaskan tentang pembahasan dari hasil pengujian yang dilakukan sebelumnya.

### 6.2.1 Pembahasan Pengujian Fungsional

Pada pengujian fungsional yang dilakukan sebelumnya, setiap fungsi yang dikembangkan pada aplikasi telah berjalan dengan baik dan mampu dioperasikan sesuai dengan hasil yang diharapkan. Fitur yang menggunakan layanan dari luar seperti google *cloud message* dan google maps untuk android dan website juga berjalan dengan baik. Dari hasil pengujian fungsional yang telah dilakukan maka aplikasi dapat disimpulkan telah berjalan dengan baik dan benar.

### **6.2.2 Pembahasan Pengujian Kompatibilitas Aplikasi Android**

Gambar yang ditampilkan pada hasil pengujian aplikasi pada android dengan versi API 19 dan 23 menunjukkan bahwa aplikasi android untuk kurir dan sales ini sudah dapat memenuhi target *range* API. Tidak ada perbedaan tampilan secara signifikan yang didapat dari pengujian aplikasi pada kedua versi API tersebut, serta fungsi dari aplikasi juga berjalan dengan baik.

## Lampiran A Test Case

Tabel 1 TC-01 *Login*

Test Case Code		TC-01			
Test Case Name		<i>Login</i>			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	<i>Login</i> dengan user yang sudah terdaftar	user = sales1	Sales dapat masuk kedalam aplikasi	Sistem memasukkan sales kedalam aplikasi	Sukses
		password= sales 1			
2	<i>Login</i> dengan user yang belum terdaftar	user = sales45	User tidak dapat masuk kedalam aplikasi	Sistem menampilkan pesan “username or password invalid”	Sukses
		password=sales45			

Tabel 2 TC-02 Tambah pesanan obat baru

Test Case Code		TC-02			
Test Case Name		Tambah pesanan obat baru			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Sales memasukkan nama obat beserta jumlahnya	Nama obat = Carboplatin	nama dan jumlah obat disimpan oleh sistem	Sistem menampilkan nama dan jumlah obat kedalam listview	Sukses
		Jumlah= 100 tablet			

Tabel 3 TC-03 Simpan lokasi pemesan

Test Case Code		TC-03			
Test Case Name		Simpan lokasi pemesan			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Sales memilih titik koordinat lokasi pemesan pada google maps	Titik koordinat dipilih pada jalan bundaran ITS	Sistem menyimpan titik koordinat lokasi yang dipilih.	Sistem menampilkan nama jalan pada form isian alamat pemesan	Sukses

Tabel 4 TC-04 Melihat detail pesanan

Test Case Code		TC-04			
Test Case Name		Melihat detail pesanan			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Sales memilih salah satu pesanan yang ada pada listview	Nama pemesan = Apotek keputih	Sistem menampilkan nama-nama obat yang dipesan beserta jumlahnya	Sistem menampilkan nama-nama obat beserta jumlahnya kedalam <i>listview</i>	Sukses

Tabel 5 TC-05 Logout

Test Case Code		TC-05			
Test Case Name		Logout			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Logout dengan sales yang sudah login	user = sales1	sales keluar dari aplikasi	Sistem menutup sesi user dan menampilkan halaman login	Sukses

Tabel 6 TC-06 *Login*

Test Case Code		TC-06			
Test Case Name		<i>Login</i>			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	<i>Login</i> dengan user yang sudah terdaftar	user = kurir1	Kurir dapat masuk kedalam aplikasi	Sistem memasukkan kurir kedalam aplikasi	Sukses
		password= kurir1			
2	<i>Login</i> dengan user yang belum terdaftar	user = kurir20	Kurir tidak dapat masuk kedalam aplikasi	Sistem menampilkan pesan “username or password invalid”	Sukses
		password=kurir20			

Tabel 7 TC-07 Melihat detail pesanan

Test Case Code		TC-07			
Test Case Name		Melihat detail pesanan			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Kurir memilih salah satu pesanan yang ada pada listview	Nama pemesan = Apotek keputih	Sistem menampilkan nama-nama obat yang dipesan beserta jumlahnya	Sistem menampilkan nama-nama obat beserta jumlahnya kedalam <i>listview</i>	Sukses



Tabel 8 TC-08 Melihat lokasi pemesanan

Test Case Code		TC-08			
Test Case Name		Melihat lokasi pemesanan			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Kurir menekan tombol “Lokasi” pada detail pesanan	Nama pemesan = Apotek keputih	Sistem menampilkan lokasi apotek keputih didalam google maps	Sistem menampilkan marker lokasi apotek didalam google maps dan secara otomatis mengatur tampilan agar fokus ke lokasi tersebut	Sukses

Tabel 9 TC-09 Mengupload foto dokumentasi

Test Case Code		TC-09			
Test Case Name		Mengupload foto dokumentasi			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Kurir menekan tombol “upload foto”	Foto diambil menggunakan aplikasi kamera	Sistem menyimpan foto dan mengubah status pesanan menjadi “selesai”	Sistem menyimpan foto dan menampilkan pesan sukses	Sukses
2	Kurir menekan tombol “cancel” ketika mengambil foto	Foto diambil menggunakan aplikasi kamera	Sistem tidak menyimpan foto dan status pesanan tidak berubah	Sistem menampilkan halaman upload foto	Sukses

Tabel 10 TC-10 *Push notification*

Test Case Code		TC-10			
Test Case Name		<i>Push notification</i>			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Kurir menekan notifikasi pesanan yang masuk	Notifikasi pesanan = Apotek Keputih	Aplikasi kurir obat dibuka langsung dan menampilkan daftar pesanan	Sistem menampilkan daftar pesanan untuk kurir tersebut	Sukses
2	Kurir menekan tombol “ <i>dismiss notification</i> ”	Notifikasi pesanan = Apotek Keputih	Aplikasi kurir obat tidak dibuka	Aplikasi tidak dijalankan	Sukses

Tabel 11 TC-11 *Track* lokasi

Test Case Code		TC-11			
Test Case Name		<i>Track</i> lokasi			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Kurir menekan tombol “track lokasi”	Titik latitude dan longitude posisi kurir	Sistem menyimpan koordinat posisi kurir setiap selang waktu 3 detik	Sistem menampilkan pesan <i>Toast</i> “ <i>Tracking location</i> ”	Sukses

Tabel 12 TC-12 *Logout*

Test Case Code		TC-12			
Test Case Name		<i>Logout</i>			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	<i>Logout</i> dengan kurir yang sudah login	user = kurir1	Sistem menutup sesi user kurir	Sistem menampilkan halaman login	Sukses

Tabel 13 TC-13 *Login*

Test Case Code		TC-13			
Test Case Name		<i>Login</i>			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	<i>Login</i> dengan user yang sudah terdaftar	user = admin	Distributor dapat masuk kedalam aplikasi	Sistem memasukkan distributor kedalam aplikasi	Sukses
		password= admin			
2	<i>Login</i> dengan user yang belum terdaftar	user = admin2	Distributor tidak dapat masuk kedalam aplikasi	Sistem menampilkan pesan “username or password invalid”	Sukses
		password=admin2			

Tabel 14 TC-14 Mengelola pesanan

Test Case Code		TC-14			
Test Case Name		Mengelola pesanan			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Distributor menekan tombol “delete” pada salah satu pesanan	Nama pemesan = Apotek Keputih	Sistem menghapus pesanan	Sistem menampilkan konfirmasi penghapusan pesanan, jika distributor menekan tombol “Ya” maka pesanan dihapus	Sukses
2	Distributor menekan tombol “Tidak” pada konfirmasi penghapusan	Nama pemesan = Apotek Keputih	Sistem tidak menghapus pesanan	Sistem menampilkan daftar pesanan	Sukses

Tabel 15 TC-15 Mengelola user

Test Case Code		TC-15			
Test Case Name		Mengelola user			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Distributor menekan tombol “delete” pada salah satu user	user = kurir2	Sistem menghapus user	Sistem menampilkan konfirmasi penghapusan user, jika distributor menekan tombol “Ya” maka user dihapus	Sukses
2	Distributor menekan tombol “Tidak” pada konfirmasi penghapusan	user = kurir2	Sistem tidak menghapus user	Sistem menampilkan daftar user	Sukses

Tabel 16 TC-16 Melihat foto dokumentasi

Test Case Code		TC-16			
Test Case Name		Melihat foto dokumentasi			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Distributor menekan tombol “foto dokumentasi” pada salah satu pesanan	Nama pemesan = Apotek Keputih	Sistem menampilkan foto dokumentasi pengiriman pesanan	Sistem menampilkan foto dokumentasi	Sukses



Tabel 17 TC-17 Melihat hasil *track* lokasi kurir

Test Case Code		TC-17			
Test Case Name		Melihat hasil <i>track</i> lokasi kurir			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	Distributor memilih tanggal <i>tracking</i> lokasi kurir	user = kurir1	Sistem menampilkan hasil <i>tracking</i> lokasi kurir	Sistem menampilkan <i>polyline</i> perubahan posisi kurir pada google maps	Sukses
2	Distributor memilih tanggal yang tidak memiliki data <i>tracking</i> lokasi kurir	user = kurir1	Sistem tidak menampilkan data <i>tracking</i>	Sistem menampilkan pesan “Data tidak ada”	Sukses

Tabel 18 TC-18 *Logout*

Test Case Code		TC-18			
Test Case Name		<i>Logout</i>			
No.	Aksi Test	Data Test	Hasil yang diharapkan	Hasil Aktual	Sukses/ Gagal
1	<i>Logout</i> dengan distributor yang sudah login	user = admin	Sistem menutup sesi user distributor	Sistem menampilkan halaman login	Sukses

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Pada bab ini dijabarkan mengenai kesimpulan yang didapat dari hasil penelitian yang telah dilakukan. Kesimpulan ini diharapkan dapat menjawab tujuan yang telah ditetapkan diawal penelitian. Saran diberikan untuk digunakan pada penelitian selanjutnya.

#### **7.1 Kesimpulan**

Dari pelaksanaan penelitian tugas akhir ini, dapat diambil kesimpulan sebagai berikut.

1. Proses pemesanan obat dan monitoring lokasi kurir dilakukan dengan memanfaatkan teknologi Google *Cloud Messaging*, Google *Maps API*, *Java*, dan *PHP*.
2. Melalui pengujian fungsionalitas diketahui bahwa semua fungsi aplikasi sudah berjalan dengan baik.
3. Aplikasi dapat berjalan dengan baik sesuai dengan target *range API* yang ditentukan yakni versi 19-23.

#### **7.2 Saran**

Tugas akhir ini memiliki potensi yang besar untuk dikembangkan kedepannya agar lebih baik dan lebih bermanfaat, untuk itu diberikan beberapa saran yang dapat dipertimbangkan yaitu:

1. Agar aplikasi dapat mendukung pada versi API dibawah versi *minimum* pada penelitian ini, maka metode sinkronisasi data dengan *Library Volley* dapat diganti dengan *Library* sejenis yang mendukung pada versi yang akan ditargetkan.

2. Perlu dibuat sebuah *user guide* untuk aplikasi android yang digunakan oleh kurir dan sales, agar aplikasi dapat digunakan sebagaimana seharusnya.
3. Metode *tracking* lokasi kurir dapat diganti jika ingin memantau posisi kurir secara *real time* dengan memanfaatkan layanan seperti pubnub data *stream network* yang dapat diakses pada website <http://pubnub.com>.

## DAFTAR PUSTAKA

- [1] S. Wijayanti dan D. Widianik, “PERANCANGAN APLIKASI WEB MOBILE PEMESANAN OBAT (STUDI KASUS DI PEDAGANG BESAR FARMASI PT. RATNA INTAN KUSUMA),” *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011 (Semantik 2011)*, 2011.
- [2] eMarketer, “Asia-Pacific Boasts More Than 1 Billion Smartphone Users,” 16 September 2015. [Online]. Available: <http://www.emarketer.com/Article/Asia-Pacific-Boasts-More-Than-1-Billion-Smartphone-Users/1012984>. [Diakses 20 Maret 2016].
- [3] I. D. Corporation, “Android and iOS Squeeze the Competition, Swelling to 96.3% of the Smartphone Operating System Market for Both 4Q14 and CY14, According to IDC,” 24 Februari 2015. [Online]. Available: <http://www.idc.com/getdoc.jsp?containerId=prUS25450615>. [Diakses 20 Maret 2016].
- [4] M. Setianna, Hendry dan Y. Nataliani, “Perancangan dan Implementasi Aplikasi Pelaporan Penjualan dan Pengecekan Stok Berbasis Android (Studi kasus: Makintech Multimedia Centre Salatiga),” Universitas Kristen Satya Wacana, Salatiga, 2012.
- [5] M. Suwastiko, M. Richard Meisel dan A. Wahyu Sudrajat, “Sistem Informasi Pemantauan Tenaga Penjual Berbasis Lokasi (Studi Kasus : PT. Surya Mustika Lampung Cabang Baturaja),” 20 Agustus 2015. [Online]. Available: <http://eprints.mdp.ac.id/id/eprint/1636>. [Diakses 10 Maret 2016].
- [6] Y. Hartini dan Sulasmono, Apotek Ulasan Beserta Naskah Peraturan perundang-undangan terkait Apotek, Yogyakarta: Universitas Sanata Dharma, 2006.

- [7] Lembaga Negara Republik Indonesia, “PERATURAN PEMERINTAH REPUBLIK INDONESIA NOMOR 51 TAHUN 2009 TENTANG PEKERJAAN KEFARMASIAN,” 2009. [Online]. Available: <http://www.sjdih.depkeu.go.id/fulltext/2009/51tahun2009pp.htm>. [Diakses 12 Maret 2016].
- [8] A. Pranindya, Pendeteksi dan Pelacakan Keberadaan Manusia Menggunakan, Politeknik Negeri Sriwijaya, 2014.
- [9] R. Andikasani, M. Awaluddin, M. Suprayogi dan A. Darmo Yuwono, APLIKASIPERSEBARAN OBJEK WISATA DI KOTA SEMARANG BERBASIS MOBILE GISMEMANFAATKAN SMARTPHONE ANDROID, Universitas Diponegoro, 2014.
- [10] S. D. A. Anindito, PEMBANGUNAN APLIKASI ANDROID UNTUK LAYANAN PESAN ANTAR BARANG ATAU JASA BERBASIS LOKASI DAN TEKNOLOGI PUSH NOTIFICATION, Yogyakarta: Universitas Atma Jaya Yogyakarta, 2012.
- [11] A. “Analisis dan Perancangan Sistem Manajemen Event Berbasis Mobile Push Notification,” *IJCCS*, vol. 5, no. 3, pp. 63-70, 2011.
- [12] D. R. A. E. “RANCANG BANGUN APLIKASI KAMUS IRREGULAR VERB,” 2012. [Online]. Available: <http://digilib.unila.ac.id/id/eprint/14757>. [Diakses 12 Maret 2016].
- [13] S. dan S. Murugaiyan, “WATEERFALL Vs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC,” *International Journal of Information Technology and Business Management*, vol. 2, no. 1, p. 27, 2012.
- [14] “Kalbe Medical Portal,” KALBE, [Online]. Available: <http://www.kalbemed.com/Products/Drugs/Generic.aspx>. [Diakses 30 Mei 2016].
- [15] A. B. Hartanto, “Aplikasi Bergerak Sistem Manajemen Distribusi Barang dan Monitoring Salesman,” 13 12

2012. [Online]. Available:  
<http://repository.uksw.edu/handle/123456789/1231>.  
[Diakses 10 Maret 2016].

## BIODATA PENULIS



Penulis lahir di Silaban, Kabupaten Humbang Hasundutan, Sumatera Utara pada tanggal 14 November 1994. Penulis merupakan anak ketiga dari empat bersaudara. Penulis menempuh pendidikan formal di SDN 1773466 Silaban Margu, SMPN 4 Lintongnihuta, dan SMA Katolik Budi Mulia Pematang Siantar. Pada tahun 2012 penulis melanjutkan pendidikan di Jurusan Sistem Informasi – Institut Teknologi Sepuluh Nopember (ITS). Selama menempuh pendidikan kuliah, penulis aktif dalam berorganisasi, dibuktikan dengan menjadi staff Divisi *Informasi Media*, Badan Eksekutif Mahasiswa – FTIf ITS. Selain itu juga mengikuti berbagai kegiatan kemahasiswaan dan kepanitiaan. Pada semester ke-delapan perkuliahan, penulis mulai mengerjakan Tugas Akhir di Laboratorium Akuisisi Data dan Diseminasi Informasi, dibawah bimbingan Bapak Faizal Johan Atletiko, S.Kom, M.T. Semoga penulisan Tugas Akhir ini mampu memberikan kontribusi positif bagi semua pihak terkait.